粗糙关系数据库

安秋生　著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书主要研究了粗糙关系数据库理论、粗糙集与关系数据库的关系以及粗糙集理论在数据库中的应用。对粗糙集与关系数据库之间的关系、粗糙关系数据库模型、粗糙关系数据库与模糊关系数据库的关系、粗糙数据查询、粗糙函数依赖及其推理机制、基于粗糙集与信息颗粒的聚类方法、信息系统函数依赖的信息颗粒原理与计算、基于粗糙集的关系数据库范式及粗糙函数依赖的近似度量等专题进行了系统的阐述，并将其应用于数据挖掘及数据查询中，反映了当前该理论的最新研究成果。

本书可以作为计算机科学、信息科学和管理工程等高年级本科生及硕士研究生的教材，对相关学科领域的研究人员和工程技术人员也有重要的使用和参考价值。

图书在版编目（CIP）数据

粗糙关系数据库 / 安秋生著. —北京：电子工业出版社，2009.5

ISBN 978-7-121-08744-8

I. 粗… Ⅱ. 安… Ⅲ. ①粗糙集—研究 ② 关系数据库—研究
Ⅳ.O144 TP311.138

中国版本图书馆 CIP 数据核字（2009）第 065954 号

责任编辑：赵 娜
印 刷：
装 订：
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：850×1168 1/32 印张：6 字数：158 千字
印 次：2009 年 5 月第 1 次印刷
印 数：2 000 册 定价：19.80 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。服务热线：（010）88258888。

作 者 简 介

安秋生，副教授，硕士生导师，山西师范大学软件研究所副所长，九三学社山西师大副主委。1987 年 7 月毕业于山西大学计算机科学系，1995 年 5 月获得西安交通大学计算机系工学硕士学位，2003 年 12 月获得西安交通大学电信学院软件所工学博士学位，2005 年 4 月—2008 年 3 月在西安交通大学数学博士后流动站从事博士后研究工作，师从张文修教授。主要从事数据库系统理论、粗糙集和粒计算等方面的研究，CRSSC2005—2008 程序委员会委员、CWI2007 程序委员会主席、《山东大学学报》自然科学版审稿专家。曾主持中国博士后基金一项（No. 200538603），山西省自然科学基金一项（No.2006011038），并参与了国家自然科学基金（No.6047 3739,60573074）。在粗糙关系数据库研究方面首次提出了粗糙关系数据库的分解算子及粗糙函数依赖冗余因子的概念，并系统地给出了利用 Fuzzy Sets 与 Rough Sets 进行聚类分析的方法。已发表学术研究论文二十余篇，其中 SCI 检索 3 篇（第一作者），EI 收录 3 篇（第一作者），ISTP 收录 3 篇。

序

经过近三十年的发展，粗糙集基本理论的研究已日趋成熟，目前对粗糙集的研究主要集中在把粗糙集理论与相关学科的结合方面。粗糙集理论是从研究信息表（也称信息系统，或知识表达系统）的逻辑特性开始的，而关系数据库理论是从研究二维表开始的，信息表或信息系统实际上是数据库关系的泛化，这表明粗糙集理论和数据库理论有一种天然的联系，因此利用粗糙集理论和技术研究并解决与数据库有关的理论与应用问题是十分有必要的。

Beaubouef Theresa Ann 博士通过对粗糙集理论和关系数据库理论的研究，于 1993 年提出把粗糙集与关系数据库相结合形成粗糙关系数据库（Rough Relational Database，RRDB），以此为基础对粗糙关系操作算子、粗糙关系数据库的不确定性度量、粗糙函数依赖、精确数据的粗糙数据查询（RQCD）、模糊关系数据库模型、函数依赖与知识发现等专题进行了初步的研究，并把它们应用于地理信息系统中。日本学者 Shoji Hirano 和 Shusaku Tsumoto 等提出了利用粗糙集原理来进行数据库聚类分析的思想。近年来随着信息颗粒与粒化计算的出现，有许多学者开始把它们用于数据挖掘与知识发现，T.Y.Lin 发表系列论文，研究了与粒计算有关的关系数据库面向机器的数据挖掘建模理论问题。

本书汇集了作者在攻读博士学位及博士后期间的主要研究成果，这些成果主要发表在“Rough Sets, Fuzzy Sets, Data Mining and Granular Computing，LNAI 2639”、“Rough Sets and Current Trends in Computing，LNAI 3066”、“Rough Sets and Knowledge Technology ，LNAI 4062”及《模式识别与人工智能》、《小型微

型计算机系统》、《西安交通大学学报》等期刊上，引起了同行的广泛关注。书中详细地介绍了粗糙关系数据库理论、粗糙集与数据库的关系以及粗糙集理论在数据库中应用的研究成果，内容丰富、文献翔实，既注重理论推导的严谨性，同时又兼顾应用。相信该书的出版将对我国在粗糙关系数据库方面的研究起到积极的推动作用。

山西大学副校长
梁吉业 教授

前　言

波兰学者 Z. Pawlak 在 20 世纪 80 年代初提出了粗糙集理论，其本质思想是利用不可分辨关系建立论域的一个划分，得到不可区分的等价类，从而建立一个近似空间来进行粒度计算，目前粗糙集已成为粒度计算的主要研究工具之一。

美国圣荷西州立大学 T.Y. Lin 教授于 1996 年向 Lotfi A.Zadeh 提出作“Granular Computing”的研究，Zadeh 称其为“Granular Mathematics”，T.Y. Lin 改称为“Granular Computing”，并缩写成 GrC。粒度计算从广义上来说是一种看待客观世界的世界观和方法论。

Beaubouef Theresa Ann 博士通过对粗糙集和关系数据库理论的研究，于 1993 年提出粗糙关系数据库，并以此为基础对粗糙关系操作算子、粗糙关系数据库的不确定性度量、粗糙函数依赖、精确数据的粗糙数据查询、模糊关系数据库模型和函数依赖与知识发现等专题进行了初步的研究，并应用于地理信息系统中。

本书在粗糙集和粒计算理论框架下，系统阐述了粗糙关系数据库操作算子、数据查询、函数依赖和近似度量等问题。全书共分 9 章。第 1 章介绍了粗糙集、粒计算及粗糙关系数据库等基本理论；第 2 章研究粗糙集与关系数据库的关系，介绍了粗糙关系数据库模型，探讨了粗糙关系数据库与模糊关系数据库的关系。第 3 章研究了粗糙数据查询；第 4 章探讨了粗糙函数依赖及其推理机制；第 5 章对基于粗糙集与信息颗粒的聚类方法进行了讨论；第 6 章研究了信息系统函数依赖的信息颗粒原理与计算；第 7 章介绍了基于粗糙集的关系数据库范式；第 8 章提出了粗糙函

数依赖的近似度量；第 9 章对本书的研究进行了总结。

借本书出版之际，我要特别感谢我的博士导师沈钧毅教授和博士后导师张文修教授，两位教授不仅在学术上给了我悉心的指导，在工作和生活诸方面也给了我无私的帮助，这里谨向两位恩师表示由衷的感谢和诚挚的敬意！

我要特别感谢刘清、梁吉业、王国胤、吴渝、李德玉、吴伟志、米据生、徐久诚、张生太及郭根龙几位教授在学业上给予的无私关怀和帮助！

另外，非常感谢博士后吕晓军、孔祥玉及安诺保险经济有限公司的副总经理姚建声先生。

特别感谢好友李晓林、董署波对我治学的鼓励和做人的帮助与启发。

对父亲、母亲和妻子唐淑美、爱女安丽璇的关心支持表示由衷的感谢。

本书的出版得到了国家自然科学基金（No.70811072）的特别资助，并得到山西省自然科学基金（No.2006011038）的部分资助，在此一并表示感谢。

本书注重系统性、严谨性和可读性。可以作为计算机科学、信息科学和管理工程等高年级本科生及硕士研究生的参考教材和教学用书，对相关学科领域的研究人员和工程技术也有重要的使用和参考价值。由于作者才疏学浅，难免有不少疏漏，恳请各位专家学者批评指正，提出宝贵意见。

安秋生

2009 年 2 月

目　录

第1章　基 本 理 论

工欲善其事，必先利其器

——孔子《论语》

1.1　粗糙集

在自然科学、社会科学与工程技术的诸多领域中，都不同程度地涉及到对不确定因素和不完备信息的处理。从实际系统中采集到的数据常常包含着不精确的甚至不完整的信息，若采用纯数学上的假设来消除或回避这种不确定性，效果往往不理想。反之，如果对这种信息进行适当地处理，常常有助于实际系统问题的解决。因此多年来研究人员们一直在努力寻找科学地处理不完整性和不确定性的有效途径。

在经典逻辑中，只有真、假值之分，而现实生活中许多含糊现象并不能简单地用真、假值来表示，因此，长期以来许多逻辑学家和哲学家致力于研究含糊概念。1904 年谓词逻辑的创始人 G.Frege 就提出了含糊（Vague）一词，他把它归结到边界线上，也就是说在全域上存在一些个体既不能在其某个子集上分类，也不能在该子集的补集上分类[1]。

Lotfi A.Zadeh 于 1965 年开拓性地提出了模糊集理论，该理论是研究和处理模糊现象的，所研究的事物的概念本身是模糊的，即一个对象是否符合这个概念难以确定，这种由于概念外延的模糊而造成的不确定性称为模糊性（Fuzziness）。

对于经典数学，人们自然而然联想到“精确”二字，精确数

学建立在集合论的基础上。在康托创立的经典集合论中，经典集合所表达概念的内涵和外延都必须是明确的，一事物要么属于某集合，要么不属于某集合，二者必居其一，不允许模棱两可。但在人们的思维中，有许多没有明确外延的概念，即模糊概念。语言上有许多模糊概念的词，例如以人的年龄为论域，“青年”、“中年”、“老年”、“非年轻人”和“非老年人”都没有明确的外延，它们之间没有明确的界限，在一定意义下是一种过渡状态[2]；或者以人的身高为论域，“高个子”、“中等身材”和“矮个子”也没有明确的外延。诸如此类的概念都是模糊概念。

控制论创始人维纳在谈人胜过任何最完善的机器时说：“人具有运用模糊概念的能力。”人脑能对模糊事物进行识别和判决，但计算机对模糊现象识别能力较差，为提高计算机识别模糊现象的能力，就需要把人们常用的模糊语言设计成机器能接受的指令和程序，以便机器能像人脑那样简洁灵活地做出相应的判断，从而提高自动识别和控制模糊现象的效率，这就推动了模糊数学的研究。

康托创立的经典集合论是经典数学的基础，它的逻辑真值是以数理逻辑为基础的。Zadeh 创立的模糊集合是模糊数学的基础，它的逻辑真值是以模糊逻辑为基础的，是对经典集合的开拓。但遗憾的是模糊集是不可计算的，即没有数学公式可以描述这一含糊概念，故无法计算出它具体包含糊元素的个数，如模糊集中的隶属函数 μ 和模糊逻辑中的算子 λ 均是如此。

20 世纪 70 年代，波兰数学家 Z.Pawlak 和一些波兰科学院、波兰华沙大学的逻辑学家们一起从事关于信息系统逻辑特性的研究，粗糙集理论就是在这种研究的基础上产生的。1982 年，Z.Pawlak 发表的经典论文 Rough Sets[3]，宣告了粗糙集理论的诞生。此后，粗糙集理论引起了许多数学家、逻辑学家和计算机研究人员的兴趣，他们在粗糙集的理论和应用方面做了大量的研究。1991 年 Z.Pawlak 的专著[4]和 1992 年的应用专著的出版，对

这一段时期内理论和实践的成果做了较好的总结，同时促进了粗糙集在各个领域的应用。此后召开的与粗糙集有关的国际会议进一步推动了粗糙集的发展，越来越多的科技人员开始了解并准备从事该领域的研究。目前，粗糙集已成为人工智能领域中一个较新的学术热点，在机器学习、知识获取、决策分析及过程控制等许多领域中都得到了广泛的应用。

由于最初关于粗糙集理论的研究论文大部分是用波兰语发表的，因此当时没有引起国际计算机、数学和人工智能等领域研究者的注意，研究地域也仅局限在东欧的一些国家，直到 20 世纪 80 年代末才逐渐引起各国学者的注意。二十多年来，由于粗糙集理论在机器学习与数据库知识发现、数据挖掘、决策支持与分析、数据库系统理论等领域的广泛应用，它的研究逐渐趋热。1992 年，第一届关于粗糙集理论的国际学术会议在波兰召开。1995 年，ACM Communication 将其列为新浮现的计算机科学的研究课题。1998 年国际 Information Sciences 杂志为粗糙集理论的研究出了一期专辑[5]。

从 1992 年开始，每年都召开以 Rough Set 为主题的国际会议，国际上成立了相应的粗糙集学术研究会，并且在 Internet 上定期发布电子公告，加速了粗糙集理论的发展与交流。由于粗糙集理论能够分析处理不精确、不协调和不完备信息，因此作为一种具有极大潜力和有效的知识获取工具受到人工智能研究者的广泛关注。目前，对应粗糙集概念，发展了粗糙代数、粗糙逻辑、粗糙关系数据库和模糊粗糙关系数据等，与其他相关理论（如模糊集，证据理论）的关系也得到了研究和证明，明确了粗糙集理论在数学上的独立地位。近年来，粗糙集不但在数学理论上不断得到完善，而且在其他研究领域中也得到了成功的应用，如机器学习、决策分析、近似推理、图像处理、医疗诊断、金融数据分析、专家系统、冲突分析、过程控制和数据库知识发现（Knowledge Discovery in Database，KDD）等领域[6]。

粗糙集理论的基本思想是通过关系数据库分类归纳形成概念和规则。在粗糙集中有两个重要概念，一是近似算子，一是约简与核心。通过上、下近似算子产生确定性规则与不确定性规则，通过约简与核心简化规则使之具有较好的泛化能力。目前成功的研究结果主要体现在具有有限属性值的关系数据库上，通过等价关系的分类以及分类对于目标的近似，实现知识发现过程[7]。换句话说，粗糙集是利用已知的知识库，将不精确和不确定的知识用已知的知识库中的知识来（近似）刻画；粗糙集理论是建立在分类的基础上，它将分类理解为在特定空间上的等价关系，而等价关系构成了对该空间的划分[5]。利用粗糙集理论进行数据分析有以下特点：（1）在数据分析过程中，只有已知的数据被处理，由用户提供的需处理的数据构成了直接的信息源。它与其他处理不确定和不精确问题的理论最显著的区别是它无需提供所处理的数据集合之外的任何先验信息，即数据以外的参数假设是不需要的，如统计学中的概率分布，Dempster-Shafer 证据理论中的基本概率赋值，模糊集理论中的隶属度等，这些信息有时并不容易得到，而粗糙集理论则避免了这些问题。（2）以粗糙集理论为支撑的粗糙数据分析技术（Rough Set Data Analysis，RSDA）是一个强大的数据分析工具，它能表达和处理不完备信息，能在保留关键信息的前提下对数据进行约简并求得知识的最小表达，能识别并判定数据之间的依赖关系以去除冗余属性从而达到降低维数的目的，能从经验数据中获取易于证实的规则知识等。（3）与已知的模糊集知识形成互补。粗糙集和模糊集分别刻画了不完备信息的两个方面：粗糙集以不可分辨关系为基础，侧重分类；模糊集基于元素对于集合的隶属程度，强调集合本身的隶属性与含糊性。从粗糙集的观点看，某些集合不能精确定义的原因是缺乏足够的论域知识，但可以用一对清晰的集合来表示。（4）粗糙集和 KDD 关系密切，它为 KDD 提供了一种新的研究方法和工具。KDD 研究的实施对象多为关系型数据库。关系表可被看作为粗

糙集理论中的信息表或决策表，这给粗糙集方法的应用带来极大的方便。（5）现实世界中的规则有确定性的，也有不确定性的。从数据库中发现不确定性的知识，为粗糙集方法提供了用武之地。另外，运用粗糙集方法得到的知识发现算法有利于并行执行，这极大地提高了对大型数据库知识发现的效率[8]。

目前，对粗糙集的研究主要集中在：粗糙集模型的推广，问题的不确定性的研究，与其他处理不确定性及模糊性问题的数学理论的关系与互补纯粹的数学理论方面的研究，粗糙集数据约简与知识获取的算法研究；粗糙集与数据库关系的研究；粗糙集与模糊集、商空间、粒计算相互之间关系研究等。这些研究有的是受应用的推动而产生的，有的是纯理论的，尚无应用背景。

在粗糙集模型的推广方面的研究主要涉及可变精确粗糙集模型、模糊粗糙集模型与粗糙模糊集模型、基于相似关系的粗糙集模型、基于一般关系的粗糙集模型、α-RST 模型、基于优先关系的粗糙集模型、不完备系统下的粗糙集模型以及对连续属性的离散化等[10~26]。

粗糙集理论中的不确定性主要由两个原因产生：来自论域上的二元关系及其产生的知识模块，即近似空间本身，如果二元等价关系产生的每一个等价类只有一个元素，那么由等价关系产生的划分不产生任何信息。论域的划分越粗糙，则每一个知识模块越大，知识库中的知识越粗糙，相对于近似空间的概念和知识就越不确定，这时处理知识的不确定性往往用 Shannon 的信息熵来刻画。从这个角度讲，粗糙集与信息论的关系就比较密切，不少学者在这方面做了研究工作[5]。

在粗糙集与其他处理模糊性及不确定性方法之间关系的研究中，主要讨论它与模糊集理论和 Dempster-Shafer 证据理论的关系和互补。

（1）具体地说，模糊集和粗糙集理论在处理不确定性和不精确性问题上都推广了经典的集合论。它们虽然有一定的相容性和

相似性，但它们对知识刻画的侧重面不同。从知识的“粒度”的描述上看，模糊集是通过对象关于集合的隶属程度来近似描述的，而粗糙集是通过一个集合关于某个可利用的知识库上的一对上、下近似来描述的；从集合对象间的关系来看，模糊集强调的是集合边界的定义，即边界的不分明性，而粗糙集强调的是对象间的不可分辨性；从研究的对象上看，模糊集研究的是属于同一类的不同对象间的隶属关系，重在隶属程度，而粗糙集研究的是不同类中的对象组成的集合关系，重在分类。虽然模糊集的隶属函数与粗糙集的粗糙隶属函数均反映了概念的模糊性，直观上有一定的相似性，但模糊集的隶属函数大多是由专家的经验给出的，因此往往带有很大的主观性，而粗糙集的粗糙隶属函数是对数据进行分析计算出来的，因此非常客观。正因如此，将粗糙集理论与模糊集理论进行某些“整合”，然后描述知识的不确定性和不精确性比它们各自描述知识的不确定性和不精确性可以显示更强的功能，模糊粗糙集模型是个比较成功的范例[5, 12]。

（2）粗糙集理论与 D-S 证据理论在处理不确定问题上产生和研究的方法是不一样的，但是却有某种相容性，粗糙集理论是为开发规则的机器自动生成而提出的，而 D-S 理论主要用于证据推理。粗糙集理论用概念的一对上、下近似来进行描述，而 D-S 证据理论则用一对信任函数和似然函数在给定证据下对假设进行评估。粗糙集理论中的下近似和上近似的概率恰好分别是信任函数和似然函数[27]，然而生成信任函数和似然函数的基本概率分配函数（即 mass 函数）是不同的，前者来自系统中的数据本身，比较客观，而后者来自专家的经验，比较主观，因此粗糙集理论与 D-S 证据理论具有很强的互补性[5]。

在粗糙集理论数学性质研究方面，主要讨论粗糙集的代数结构与拓扑结构，以及粗糙集的收敛性问题[28~31]。一些衍生数学概念也不断出现，如粗糙理想、粗糙半群[32]和粗糙群等。相信随着粗糙结构、拓扑结构和序结构等各种结构的不断整合，必将涌现

出新的富有生机的数学分支。

在粗糙集理论用于数据约简与知识获取的有效性算法研究方面，主要集中于抽取最优决策规则算法[33]、导出规则的增量式算法[34~35]、约简的启发式算法[36~37]及粗糙集基本运算的并行算法[38~39]。

在粗糙集与数据库关系的研究方面，主要研究粗糙集理论与关系数据库的关系问题[40]和信息系统（数据库关系的泛化）的粗糙计算问题（包括利用属性进行分类问题、函数依赖问题、寻找数据库的键值、属性及属性子集的重要性问题）等[41]。由粗糙集与关系数据库结合形成了粗糙关系数据库（Rough Relational Database，RRDB），在这方面主要研究粗糙关系操作算子、粗糙关系数据库的不确定性的信息论度量，粗糙函数依赖及粗糙范式和粗糙数据查询等[42~43]。

在粗糙集理论中的度量方面主要研究粗糙数据分析中的度量、知识的不确定性度量及粗糙集与粗糙关系数据库的信息度量[4, 26, 43]。另外，由于基于粗糙集的逻辑是关于粗糙集的不确定推理的基础，发展这类逻辑理论基础也是目前粗糙集理论研究的重要课题。

粗糙集理论的生命力在于它具有较强的实用性，从诞生到现在已在许多领域取得了令人振奋的成果[6~8]：

（1）在股票数据分析方面，文献[44]应用粗糙集方法分析了十年间股票的历史数据，研究了股票价格与经济指数之间的依赖关系，获得的预测规则得到了华尔街证券交易专家的认可。

（2）在医疗诊断方面，粗糙集方法根据以往的病例归纳出诊断规则，用来指导新的病例。现有的人工预测早产的准确率只有17%～38%，利用粗糙集方法则可以提高到 68%～90%[45~46]。

（3）在地震预报方面，文献[47]研究了地震前的地质和气象数据与里氏地震级别的依赖关系，为地震的预测提供了一种新的方法。

（4）在模式识别方面，文献[48]应用粗糙集方法研究了手写字符识别问题，提取出了相应的特征属性。

（5）决策分析[49~50]。基于粗糙集的决策规则是在分析以往经验数据的基础上得到的。粗糙集允许决策对象中存在一些不太明确和不完整的属性，弥补了常规决策方法的不足。希腊工业发银行 ETEVA 应用粗糙集理论协助制定信贷政策，是粗糙集多准则决策方法的一个成功范例[51]。

（6）专家系统。粗糙集抽取决策规则的特点为构造专家系统的知识库提供了一条崭新的途径[46]。

（7）人工神经网络。训练时间过于漫长的固有缺点是制约神经网络实用化的因素之一。文献[37]应用粗糙集约简神经网络训练样本数据集，使训练速度提高了 4.77 倍，获得了较好的效果。文献[52～53]将粗糙集与神经网络结合起来，充分利用粗糙集处理不确定性的特长以增强神经网络的信息处理能力。

（8）粗糙控制。粗糙集根据观测数据获得控制策略的方法被称为从范例中学习，属于智能控制的范畴。基本步骤是：把控制过程中的一些代表性的状态以及操作人员在这些状态下所采取的控制策略记录下来，形成决策表，然后对其进行分析约简，总结出控制规则。文献[54～55]应用粗糙控制研究小车—倒立摆系统这一经典问题，取得了较好的效果。在过程控制领域，文献[56]应用粗糙集方法成功提取了水泥窑炉的控制规则。粗糙控制的优点是简单迅速、容易实现，不需要像模糊控制那样进行模糊化。因此在特别要求控制器结构与算法的场合，采取粗糙控制较为合适。另外，由于控制算法完全来自观测数据本身，其决策和推理过程可以很容易被检验和证实。一种新的有吸引力的控制策略——粗糙控制策略正在悄然兴起，其主要思想是利用粗糙集获取模糊控制规则。

（9）冲突分析。文献[57]应用粗糙集方法建立了反映以色列、巴勒斯坦、约旦、埃及、叙利亚和沙特阿拉伯六国关于中东

和平问题各自立场的谈判模型。

（10）数据库知识发现。KDD 是当前人工智能和数据库技术交叉学科的研究热点之一。粗糙集方法现已成为 KDD 的一种重要方法，其导出的知识精练且更便于存储使用。与其他知识发现方法相比，粗糙集方法有如下特点：粗糙集方法的伸缩性强；鲁棒性和抗噪声能力强；知识的可理解性和开放性较好；比较适合于符号信息。此外，粗糙集方法可以对数据进行预处理，去掉多余属性，提高发现效率，降低错误率。

王珏教授在谈到 Rough Set 理论对归纳机器学习的贡献时认为，基于 Rough Sets 的 Reduct（约简）理论在归纳机器学习理论中起着重要的作用，而 Roughness 也成为粒度计算中对知识粒度的一个重要测量。他把粗糙集的贡献总结为三点：①规范归纳机器学习，Rough Sets 可以作为归纳机器学习的理论基础：Rough Set 理论第一次揭示了归纳机器学习中的两个模型 ID3 和 AQ11 的关系和本质；②独立约简，这是一个有数学基础的结构目标，从而部分解决了机器学习平凡性的问题：与经典的 ID3 和 AQ11 之类的归纳机器学习相比，Rough Set 理论所暗示的研究空间要大得多。由于独立约简是一个介于最优与完全随意（事实上就是无目标）之间的一个问题复杂性 $O(n^2)$的目标，对机器学习而言，其重要性就不仅限于结构化机器学习的研究了，它已具有了更为深刻的含义；③正区域与 Roughness，以描述知识粒度与其测量：Roughness 的提出使得信息系统的知识粒度可以使用 Roughness 来测量，Roughness 成为知识粒度的特征，这就是 Rough Set 理论的另外一个重要贡献[58]。

商空间理论的创始人之一张铃教授认为，粗糙集从本质上看是微观的粒度计算。研究粒度的表示、刻画和粒度与概念之间的依存关系，粗糙集理论中的论域只是简单的点集，元素之间没有拓扑关系。故在某种意义下，粗糙集理论是一种无结构的商空间理论。

虽然粗糙集理论至今只有近三十年的发展历史，但相关的研究成果是令人注目的，它是一种非常有前途的软计算方法，为处理不确定性信息提供了强有力的分析手段，相信粗糙集理论具有广阔的发展空间，今后将会在更多的领域发挥作用。

同时也应指出，在数据挖掘技术与理论非常丰富的今天，粗糙集理论也不是万能的。对建模而言，尽管粗糙集方法对知识不完全处理是有效的，但是由于未包含处理不精确或不确定原始数据的机制，因此，单纯地使用这个理论不一定能有效地描述与处理不精确或不确定问题，这意味着需要其他方法补充。粗糙集理论与其他软计算方法的结合能够提高数据挖掘能力，这是由现实世界的复杂性和处理方法有限能力的矛盾决定的。因此，在处理不确定问题时，往往将粗糙集方法与其他方法结合，如粗糙集与模糊集、证据理论、神经网络、遗传算法、信息论方法、统计方法、Petri 网和 Bayesian 方法等相结合，可以说，粗糙集与其他软计算方法的结合是粗糙集发展的一种趋势。

1.1.1 信息系统

任何一种智能信息处理方法都不可能离开适合于其自身的知识表达形式，基于粗糙集理论的知识发现也不例外，它主要是借助于信息系统来展开其理论和方法的。

从认知科学的角度来看，可以认为知识是人类对客观事物的分类能力，概念为事物类的描述。一般地，人们在研究问题的时候总是在感兴趣的事物范围内来进行的，该范围即数学上所说的论域。假定起初对论域中的个体（对象）具有必要的信息或概念，这些信息和概念相对于要发现的知识都是原始的和基本的。通过这些概念可以将论域中的对象划分到不同的类别。如果两个对象具有完全相同的信息，那么将无法分辨它们，或者称它们是不可分辨的。按照论域中的对象之间是否具有不可分辨性，就可以在论域上建立对象之间的一个二元关系，显然这是一个等价

关系[5~6]。

集合上的等价关系和集合的划分是等价的概念，即集合上的等价关系和其上的划分是可以相互唯一决定的。而划分就是分类，所以等价关系与分类具有天然的联系。论域中由等价关系划分出来的任意子集，都可称为论域中的一个概念。通常把论域中的任意概念族称为关于论域的抽象知识，简称为知识，它也代表了对论域中对象的分类[4]。

设 U 是一个论域，R 是其上的一个等价关系族，根据上面的讨论，知识可以形式化地定义为在等价关系族 R 下对论域 U 中对象的划分，记作 U/R。信息系统是粗糙集理论对知识进行表达和处理的基本工具。

定义 1.1[4]　信息系统是一个有序四元组 $S=(U, \mathrm{AT}, V, f)$，其中，

U——称为论域，由对象组成，它是一个有限非空集合；

AT——称为属性集，它是一个有限的非空属性集合；

V_a——称为属性值域，V_a 是属性 a 的值域；

f——是一个 $U\times \mathrm{AT}\rightarrow V$ 的映射，对 $\forall u\in U$，$a\in \mathrm{AT}$，$f(u, a)\in V_a$，通常称 f 为信息函数或描述函数。

事实上，信息系统可直观地表达为一个二维表的形式，通常称该二维表为信息表，它是表达描述知识的数据表格.

定义 1.2[4]　一般地，定义 1.1 中的 $\mathrm{AT}=C\cup D$，C 称为条件属性，D 称为决策属性，如果 $D=\phi$，此时的信息系统即为一般的信息系统；如果 $D\neq\phi$，则该信息系统称为决策信息系统，表达决策信息系统的信息表称为决策表。

1.1.2　近似集及其性质

在信息系统中，对一个概念（即论域的一个子集）进行刻画时，一般只能通过信息系统的基本概念来解释，但由于现有的信息不足，并非对所有的概念都能精确地刻画和描述，也就是说，

此时必须面对用基本概念近似描述任意概念的任务。粗糙集理论中的近似空间正是基于这种要求而提出的[8]。

定义 1.3[4] 称论域 U 连同其上定义的一个二元关系 R 所形成的有序二元组 $S=(U, R)$为论域 U 上的一个近似空间。

论域上一个二元关系可以按照某种方式决定论域的一个划分或覆盖，而这些划分或覆盖中的成员将被看作基本概念，用来描述一般的概念。特别地，如果关系 R 是一个等价关系（在粗糙集理论中也称为不可分辨关系（Indiscernibility Relation）），则它唯一地决定了论域的一个划分。

令$[x]_R=\{y\in U|(x, y)\in R\}$，称$[x]_R$为由 R 决定的 x 的等价类，关系 R 的等价类称为 S 中的基本集（基本概念）或原子。

S 中任何有限基本集的并称为 S 中的可定义集。S 中所有可定义集用符号 Def(S)来表示。可定义集反映的是论域中可以被基本概念精确描述的概念。

知识的粒度性是造成已有知识不能精确地表示某些概念的原因。这就产生了关于不精确的“边界”思想。著名哲学家 G.Frege 认为“概念必须有明确的边界，没有明确边界的概念，将对应一个在周围没有明确界线的区域。”粗糙集中的模糊性就是一种基于边界的概念，即一个不精确的概念具有模糊的不可被明确划分的边界，为刻画模糊性，每个不精确概念用一对称为上近似与下近似的精确概念来表示[1]。

设 $X\subseteq U$，集合 X 关于 R 的下近似（Lower Approximation）定义为：

$$\underline{apr}_{\mathrm{R}}(X)=\{x\in U|[x]_R\subseteq X\} \qquad (1.1)$$

$\underline{apr}_{\mathrm{R}}(X)$ 实际上是由那些根据已有知识判断肯定属于 X 的对象所组成的最大集合，也称为 X 的正区域（Positive Region），记作 $\mathrm{POS}_{\mathrm{R}}(X)$。

由根据已有知识判断肯定不属于概念 X 的对象所组成的集合

称为 X 的负区域（Negative Region）。记作 $\mathrm{NEG}_R(X)$：

$$\mathrm{NEG}_R(X)=\{x\in U|[x]_R\cap X=\phi\}$$

集合 X 关于 R 的上近似（Upper Approximation）定义为：

$$\overline{apr}_R(X)=\{x\in U|[x]_R\cap X\neq\phi\} \tag{1.2}$$

$\overline{apr}_R(X)$是由那些根据已有知识判断可能属于 X 的对象所组成的最小集合。显然，$\overline{apr}_R(X)\cup\mathrm{NEG}_R(X)=U$。

集合 X 关于 R 的边界区域（Boundary Region）定义为：

$$\mathrm{BN}_R(X)=\overline{apr}_R(X)-apr_R^{(X)} \tag{1.3}$$

$\mathrm{BN}_R(X)$为集合的上近似与下近似之差。如果 $\mathrm{BN}_R(X)=\phi$，则称 X 关于 R 是清晰的（Crisp），X 为精确集、可定义集；反之，如果 $\mathrm{BN}_R(X)\neq\phi$，则称 X 为关于 R 的粗糙集。

我们也可以将 $\underline{apr}_R(X)$看作为 X 中的最大可定义集，将 $\overline{apr}_R(X)$看作为含有 X 的最小可定义集。

下近似和上近似具有下列性质。

定理 1.1[4]　$\underline{apr}_R(X)$和 $\overline{apr}_R(X)$有下列性质：

（1）$\underline{apr}_R(X)\subseteq X\subseteq\overline{apr}_R(X)$；

（2）$\underline{apr}_R(\phi)=\overline{apr}_R(\phi)=\phi$；

（3）$\underline{apr}_R(U)=\overline{apr}_R(U)=U$；

（4）$\overline{apr}_R(X\cup Y)=\overline{apr}_R(X)\cup\overline{apr}_R(Y)$；

（5）$\underline{apr}_R(X\cap Y)=\underline{apr}_R(X)\cap\underline{apr}_R(Y)$；

（6）$X\subseteq Y\Rightarrow\underline{apr}_R(X)\subseteq\underline{apr}_R(Y)$；

（7）$X\subseteq Y\Rightarrow\overline{apr}_R(X)\subseteq\overline{apr}_R(Y)$；

（8）$\underline{apr}_R(X\cup Y)\supseteq\underline{apr}_R(X)\cup\underline{apr}_R(Y)$；

（9）$\overline{apr}_R(X\cap Y)\subseteq\overline{apr}_R(X)\cap\overline{apr}_R(Y)$；

（10）$\underline{apr}_R(-X)=-\overline{apr}_R(X)$；

（11）$\overline{apr}_{\mathrm{R}}(-X)=-\underline{apr}_{\mathrm{R}}(X)$；

（12）$\underline{apr}_{\mathrm{R}}(\underline{apr}_{\mathrm{R}}(X))=\overline{apr}_{\mathrm{R}}(\underline{apr}_{\mathrm{R}}(X))=\underline{apr}_{\mathrm{R}}(X)$；

（13）$\overline{apr}_{\mathrm{R}}(\overline{apr}_{\mathrm{R}}(X))=\underline{apr}_{\mathrm{R}}(\overline{apr}_{\mathrm{R}}(X))=\overline{apr}_{\mathrm{R}}(X)$。

1.1.3 近似质量的刻画

粗糙集理论中，集合的不精确性是由于边界区域的存在而引起的。集合的边界区域越大，其精确性越低。为了更准确地表达集合的精确性，引入近似精度的概念。由等价关系 R 定义的集合 X 的近似精度为：

$$\alpha_{\mathrm{R}}(X)=\frac{card(\underline{apr}_{\mathrm{R}}(X))}{card(\overline{apr}_{\mathrm{R}}(X))} \tag{1.4}$$

其中 $X\neq\phi$，$card(X)$ 表示集合 X 的基数。近似精度$\alpha_R(X)$用来反映我们了解集合 X 的知识的完全程度。

显然，对每个 R 和 $X\subseteq U$ 有 $0\leqslant\alpha_R(X)\leqslant 1$。

当$\alpha_R(X)=1$ 时，X 的 R 边界区域为空集，集合 X 为 R 可定义的；

当$\alpha_R(X)<1$ 时，集合 X 有非空 R 边界区域，集合 X 为 R 不可定义的，即粗糙的[4]。

也可以用其他一些度量来刻画集合 X 的不精确程度。例如，用 X 的 R 粗糙度$\rho_R(X)$来刻画：

$$\rho_R(X)=1-\alpha_R(X) \tag{1.5}$$

X 的 R 粗糙度与近似精度恰恰相反，它表示的是集合 X 的知识的不完全程度。

也可根据上近似和下近似的定义来表达粗糙集的另一个有用的特征，即拓扑特征。下面定义四种不同的重要粗糙集[4]：

（1）如果 $\underline{apr}_{\mathrm{R}}(X)\neq\phi$且 $\overline{apr}_{\mathrm{R}}(X)\neq U$，则称 X 为 R 粗糙可定义的。

（2）如果 $\underline{apr}_{\mathrm{R}}(X)=\phi$ 且 $\overline{apr}_{\mathrm{R}}(X)\neq U$，则称 X 为 R 内不可定义的。

（3）如果 $\underline{apr}_{\mathrm{R}}(X)\neq\phi$ 且 $\overline{apr}_{\mathrm{R}}(X)= U$，则称 X 为 R 外不可定义的。

（4）如果 $\underline{apr}_{\mathrm{R}}(X)=\phi$ 且 $\overline{apr}_{\mathrm{R}}(X)=U$，则称 X 为 R 全不可定义的。

上述定义说明如果集合 X 为 R 粗糙可定义，则可以确定 U 中某些元素是否属于 X 或 $-X$；如果集合 X 为 R 内不可定义，则意味着可以确定 U 中某些元素是否属于 $-X$，但不能确定 U 中任一元素是否属于 X；如果集合 X 为 R 外不可定义，则可以确定 U 中某些元素是否属于 X，但不能确定 U 中任一元素是否属于 $-X$；如果集合 X 为 R 全不可定义，则意味着不能确定 U 中任一元素是否属于 X 或 $-X$。

前面介绍了两种刻画粗糙集的方法。一种为用近似程度的精度来表示粗糙集的数字特征；另一种为用粗糙集的分类表示粗糙集的拓扑特征。粗糙集的数字特征表示了集合边界区域的大小，但没有说明边界区域的结构；而粗糙集的拓扑特征没有给出边界区域大小的信息，它提供的是边界区域的结构。

粗糙集理论中定义了粗糙隶属函数（Rough Membership Function）。通过使用不可分辨关系，定义元素 x 对集合 X 的粗糙隶属函数如下：

$$\mu_{\mathrm{X}}^{\mathrm{R}}(x)=\frac{card(X\cap[x]_{\mathrm{R}})}{card([x]_{\mathrm{R}})} \tag{1.6}$$

显然，$0\leqslant\mu_{X}^{R}(x)\leqslant 1$。

定理 1.2[60]　粗糙隶属函数有以下性质:

（1）$\mu_{\mathrm{X}}^{\mathrm{R}}(x)=1$，当且仅当 $x\in\underline{apr}_{\mathrm{R}}(X)$；

（2）$\mu_{\mathrm{X}}^{\mathrm{R}}(x)=0$，当且仅当 $x\in-\underline{apr}_{\mathrm{R}}(X)$；

（3）$0<\mu_X^R(x)<1$ 当且仅当 $x\in BN_R(X)$；

（4）如果 $R=\{(x, x)|x\in U\}$，即 R 是 U 上的恒等关系，则 $\mu_X^R(x)$是 X 的特征函数；

（5）如果$(x, y)\in R$，则 $\mu_X^R(x)=\mu_X^R(y)$；

（6）$\mu_{U-X}^R(x)=1-\mu_X^R(x), \forall x\in U$；

（7）$\mu_{X\cup Y}^R(x)\geqslant\max(\mu_X^R(x), \mu_Y^R(x)), \forall x\in U$；

（8）$\mu_{X\cap R}^R(x)\leqslant\min(\mu_X^R(x), \mu_Y^R(x)), \forall x\in U$；

（9）如果 $X=\{X_1, X_2, \cdots, X_n\}$是 U 的一族互不相交的子集，则 $\mu_{\cup X_i}^R(x)=\Sigma_{X_i\in X}\mu_X^R, \forall x\in U$。

1.1.4 知识约简与依赖性

知识约简是粗糙集理论的重要内容之一，在信息系统 $S=(U, AT, V, f)$中，一般属性集 AT 往往不止含有一个属性，每一个属性 $a\in AT$ 按照对象在其下的取值是否相同就可以决定论域 U 上的一个等价关系，记为 IND(a)。同样，一组属性按照元素在该组属性下的取值是否均相同，也可以决定论域 U 上的一个等价关系。这些等价关系均各自决定了论域的划分。然而从形成对论域的划分不变这一角度来讲，并非所有的属性均是必不可少的。换句话说，在保持对论域分类能力不变的前提下，有些属性是多余的，删除冗余属性就是信息系统中的属性约简问题[4~7]。

设 $S=(U, AT, V, f)$是一个信息系统，对任一子集 $P\subseteq AT$，定义 U 上的二元关系 IND(P)如下：

$$IND(P)=\{(x, y)\in U\times U|\forall a\in P, a(x)=a(y)\} \quad (1.7)$$

容易验证 IND(P)是 U 上的等价关系。当 P 为单元集$\{a\}$时，IND(P)简记为 IND(a)。如果$(x, y)\in IND(P)$，则称 x 和 y 是 P 不可分辨的[4]。

用 $U/IND(P)$来表示 U 的一个划分，其中的任何元素$[x]_P$称为等价类，这里$[x]_P=\{y\in U:(x, y)\in IND(P)\}$。由于属性子集 $P\subseteq AT$ 对

IND(P)的唯一决定性，所以常用 U/P 来替代 U/IND(P)[4]。

设 P 是一个属性子集，$p\in P$ 是一个属性，如果 IND(P)=IND($P-\{p\}$)，则称 p 在 P 中是不必要的，否则称为是必要的；如果每一个 $p\in P$ 均在 P 中是必要的，则称 P 是独立的，否则称 P 是依赖的，显然，一个独立属性集的任一子集也是独立的。

设 $Q\subset P$，如果 Q 是独立的，且 IND(Q)=IND(P)，则称 Q 为 P 的一个约简。一般来讲，一个属性子集 P 可以有多个约简，P 的所有约简所成的集合记为 $red(P)$，一个属性子集 P 的所有必要的属性组成的集合称为它的核，记作 $core(P)$[4]。

对一个属性集 P，有：

定理 1.3[4]　$core(P)=\cap red(P)$.

该定理表明：一组属性的核是该组属性所表达知识的不可消去的知识特征集合；另外它还可以作为约简计算的基础。

一般地，设 P 和 Q 是论域 U 上的两个等价关系，Q 的 P 正域，记为 $pos_P(Q)$，被定义为：

$$pos_P(Q)=\cup_{X\in U/Q}\ \underline{apr}_P(X) \qquad (1.8)$$

可以看出，Q 的 P 正域是 U 中所有根据分类 U/P 的信息可以准确地被划分到关系 Q 的等价类中去的对象的集合。

在决策信息系统中，条件属性和决策属性按照它们所决定的等价关系分别把论域作了两个分类，当要用条件属性决定决策属性，特别是要得到能最大程度地表达论域对象的确定性决策规则时，所要考察的正是哪些对象根据条件属性所提供的信息可以被准确地划分到决策属性为论域所提供的分类中去，这恰好就是一个分类相对于另一个分类的正域的概念[6~7]。

但从一组条件属性中去掉一些属性时将会导致属性对论域划分的改变，此时划分会变粗，即信息的粒度会变大。这种变化就会导致决策属性在其下的正域的改变，从而改变原系统中所隐含的决策知识。但是这种分析并非对去掉任一条件属性都适合。相对约简即是在保证条件属性对决策属性的正域不变的条件下，从

条件属性组中删除多余的属性。为此有：

令 P 和 Q 为等价关系族，$p\in P$，如果

$$pos_P(Q)=pos_{P-\{p\}}(Q) \tag{1.9}$$

则称 p 为 P 中 Q 不必要的；否则 p 被称为 P 中 Q 必要的。如果 P 中的每一个属性都是 Q 必要的，则称 P 为 Q 独立的。

设 $S\subseteq P$，S 为 P 的 Q 约简，当且仅当 S 是 P 的 Q 独立子族，且 $pos_S(Q)=pos_P(Q)$，P 的 Q 约简称为相对约简。用 $red_Q(P)$来表示所有 P 的 Q 约简所成的集合。P 中所有 Q 必要的属性构成的集合称为 P 的 Q 核，简称为相对核，记为 $core_Q(P)$[4]。

类似于约简和核，相对约简和相对核有如下关系。

定理 1.4[4]　$core_Q(P)=\cap red_Q(P)$。

设 $S=(U, AT, V, f)$是一个信息系统，P，$Q\subseteq AT$ 是两个属性子集，称知识 Q 依赖于知识 P(记作 $P\Rightarrow Q$)，当且仅当 $IND(P)\subseteq IND(Q)$；称知识 P 等价于知识 Q(记作 $P\Leftrightarrow Q$)，当且仅当 $P\Rightarrow Q$ 且 $Q\Rightarrow P$；称知识 P 与知识 Q 独立，当且仅当 $P\Rightarrow Q$ 且 $Q\Rightarrow P$ 均不成立[4]。

对于属性子集 P 和 Q，尽管它们之间可能不存在依赖关系，但就其所定义的某些概念而言却存在依赖关系，即一个（或几个）概念是另一个（或几个）概念的子概念，这就出现了知识的部分依赖。

部分依赖在决策规则的刻画方面起着重要作用，它可以决定决策表中的默认规则。既然是部分依赖就有一个依赖程度的问题，如果部分依赖的依赖度很低，则这种依赖的意义就很小，反之意义就很大。知识的依赖度可以由一个分类相对于另一个分类的正域的概念来刻画。

令 $k=\gamma_P(Q)=card(pos_P(Q))/card(U)$，

称知识 Q 是 k 度依赖于知识 P 的，记作 $P\Rightarrow_k Q$。

显然，$0\leqslant k\leqslant 1$，当 $k=1$ 时，Q 完全依赖于 P；

当 $k=0$ 时，Q 完全独立于 P。

由依赖度的定义可以知道，当 $P \Rightarrow_k Q$ 时，由 Q 导出的论域的分类 $U/\mathrm{IND}(Q)$的正域覆盖了论域的 $k\times100\%$的对象。这意味着，如果希望用知识 P 来描述知识 Q，那么论域中有 $k\times100\%$的对象满足（适合）这种描述[4~7]。

知识依赖度 $k=\gamma_P(Q)=card(pos_P(Q))/card(U)$是从整体角度刻画了一个知识依赖于另一个知识的程度。它不能反映这种依赖在决策知识的概念类中的分布情况。为此用$\gamma_P(X)=card(\underline{apr}_P(X))/card(X)$，$X\in U/\mathrm{IND}(Q)$来刻画通过知识 P 能在多大程度上将知识 Q 的概念 X 中的对象进行正确划分[4~7]。

上述的$\gamma_P(Q)$和$\gamma_P(X)$给出了知识 P 关于知识 Q 的分类能力的全部信息。

通过推导，可得下列性质。

定理 1.5[4]　下列条件是等价的：

（1）$P \Rightarrow Q$；

（2）$\mathrm{IND}(P\cup Q)=\mathrm{IND}(P)$；

（3）对于所有 $X\in U/\mathrm{IND}(Q)$，有 $\underline{apr}_P(X)=X$。

定理 1.6[4]　对知识依赖有下面的性质：

（1）如果 $P\Rightarrow Q$ 且 $Q\Rightarrow R$，则 $P\Rightarrow R$；

（2）如果 $P\Rightarrow R$ 且 $Q\Rightarrow R$，则 $P\cup Q\Rightarrow R$；

（3）如果 $P\Rightarrow R\cup Q$，则 $P\Rightarrow R$ 且 $P\Rightarrow Q$；

（4）如果 $P\Rightarrow Q$ 且 $Q\cup R\Rightarrow T$，则 $P\cup R\Rightarrow T$；

（5）如果 $P\Rightarrow Q$ 且 $R\Rightarrow T$，则 $P\cup R\Rightarrow Q\cup T$；

（6）如果 $P\Rightarrow Q$ 且 $P'\supseteq P$，则 $P'\Rightarrow Q$；

（7）如果 $P\Rightarrow Q$ 且 $Q'\subseteq Q$，则 $P\Rightarrow Q'$。

1.2　粒计算

近年来，在以模糊集为基础的模糊计算和以粗糙集为基础的粗糙计算的基础上，世界各国的学者又发展了粒计算（Granular

Computing，GrC），它是一种新的软计算方法。

Lotfi A.Zadeh 于 1979 年研究了模糊集与信息粒度（Information Granularity）的关系问题[61]。Stanford 大学教授 J.R.Hobss 于 1985 年发表于在美国 Los Angeles 举行的国际人工智能联合会议上的论文“Granularity”，直接用粒度这个词作论文题目。

20 世纪 90 年代中期，随着模糊集与粗糙集理论的深入研究，信息颗粒（Information Granules）和粒计算的研究开始兴起，波兰科学家 Andrzej Skowron 于 1992 年研究了信息颗粒的规则抽取方法，随后 Andrzej Skowron 及其合作者又提出了信息颗粒的构造方法，对信息颗粒与近似空间的关系，信息颗粒的计算和粗糙公理方法等问题进行了深入的探讨[62~66]；Lotfi A.Zadeh 又于 20 世纪 90 年代中后期先后发表文章研究了模糊图、粗糙集与信息粒度的关系，探讨了信息粒度化、模糊逻辑与粗糙集的关系以及模糊逻辑与词计算的关系，分析了模糊信息颗粒化（Fuzzy Information Granulation）及其在人工推理（Human Reasoning）与模糊逻辑（Fuzzy Logic）中的理论与应用等问题。在此期间，Zadeh 提出人类认知的三个主要概念，即粒度（Granulation，包括将全体分解为部分）、组织（Organization，包括从部分集成全体）和因果（Causation，包括因果的关联），并进一步研究了粒计算。他认为，粒计算是一把大伞，它覆盖了所有有关粒度的理论、方法论、技术和工具的研究。他指出“粗略地说，粒计算是模糊信息粒度理论的超集，而粗糙集理论和区间计算是粒度数学的子集。”[67~70]

T.Y.Lin 教授于 1996 年在 UC-Berkeley 大学 Zadeh 的重点实验室做客座教授时，向 Zadeh 提出作“Granular Computing”课题的研究。当时 Zadeh 称“Granular Mathematics”，T.Y.Lin 改称“Granular Computing”后，立即得到 Zadeh 的认可，并且缩写成 GrC。所以“Granular Computing”成为今天的一个热门研究领域。T.Y.Lin 发表系列论文研究了二元关系的粒计算、粒化

模糊集、利用粒化方法进行关联规则的挖掘，以及与信息颗粒、颗粒计算有关的关系数据库的面向机器的数据挖掘建模理论问题[71~77]，提出了利用信息颗粒的位表示来进行数据挖掘的思想，他主要的工作是把该思想用于真实世界的建模、数据挖掘的建模和各种关联规则的发现，其研究方法为数据挖掘提供了一种新的思路。

他们的工作激起了学术界对粒度计算研究的兴趣，加拿大学者 Y.Y Yao 也先后发表论文研究了分层的粗糙集与粒计算、粗糙集与邻近系统（Neighborhood Systems）和粒计算的关系、信息粒化与粗糙集近似、信息表的粒计算、使用粒化计算进行数据挖掘建模等专题[78~86]，并将其应用于数据挖掘等领域。其工作要点是，用决策逻辑语言（DL-语言）来描述集合的粒度（用满足公式 f 元素的集合来定义等价类 $m(f)$），建立概念之间的 if-then 关系与粒度集合之间的包含关系的联系，并提出利用由所有划分构成的格来求解一致分类问题。所有这些研究都为知识挖掘提供了新的方法和角度。

粒计算是一门飞速发展的新学科，它融合了粗糙集、模糊集及人工智能等多种学科的研究成果。人们对粒计算的描述是建立在对它的知觉认识上的：粒计算是研究基于多层次结构的思维方式、问题求解方法、信息处理模式及其相关理论、技术和工具的学科[87]。

在我国，粒计算的研究已引起众多学者的关注与兴趣。其研究论题包括：基于商空间理论的粒计算模型、模糊商空间及粒计算的商闭包空间模型；粒计算的覆盖模糊、粗糙集与粒计算的交叉问题的研究；粒、规则与例外的关系；粒计算的理论、模型与方法的探讨；基于 Dempster-Shafer 证据理论和粗糙集的近似和知识约简；几种基于覆盖粗糙集的粒计算模型；粒逻辑及其归结原理；基于关系的粒计算模型，粒计算在进化计算、机器学习中的应和使用粒计算进行知识获取的方法；基于泛系理论的粒计算模

型；使用粒分析来描述和刻画粒计算的思考等[87]。

粒计算借助于其他学科的哲学思想和方法论，并将它们抽象成为与具体领域无关的方法和策略。它的独特性体现在用系统和结构化的理解和方法来解决复杂问题。对复杂问题的全面理解通常是多视角的，从每一个视角着眼的理解又是多层次的。由此可以得出，粒计算的过程就是对复杂问题的求解过程。它的结果表现为一个多视角和多层次的粒结构。这个粒结构是对此复杂问题的系统且近似的描述和解答。

对粒计算的研究应该着眼于三个观点：粒计算的哲学思想观点、方法论观点及计算模式观点。从哲学思想观点考虑，粒计算试图将人类的认知方式抽象化和形式化，从而提炼出结构化的思维模式；从方法论观点考虑，粒计算着重研究系统化的方法和技术，将问题求解的过程规范为结构化的、自上而下的逐步求精过程；从计算模式观点考虑，粒计算关注结构化的信息处理。粒计算的三个观点可以用三角形来表示，也可以用层次结构或三维空间模型来描述。

1.2.1 信息粒

按照 Lotfi A.Zadeh 的观点，信息粒（Information Granule）是通过不可分辨性（Indistinguish Ability）、相似性（Similarity）、近似性（Proximity）或功能性（Functionality）等来划分的对象的集合。一般来说，人类的认知的有三个基本的概念：粒度（Granulation，也称粒化）、组织（Organization）和因果（Causation），粒化把整体分解为部分，组织则是把部分集成为整体，而原因则涉及事物之间的关联[70]。

“Granule”被解释为中文词意“基本粒”，在英汉词典中被说成是紧紧凝结在一起的“颗粒”和“块”等，“Information Granules”是研究将信息集切割成互不相交的“片”和“块”等，或划分成互不相交的“子集”、“组”、“类”和“群”等，实

质上是“划分”（Partition）的意义，表示颗粒之间是清晰、互不相交的。可见粒计算是研究信息划分的。

“Granulation”被译为“粒”。Lotfi A.Zadeh 于 1997 年发表的论文：“Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzylogic ”中用了“Granulation”，“Information Granulation”意思是将信息切割或分成可能两两有交的“类”和“块”等，他从模糊集观点讨论，所以被分成的颗粒可能是模糊或不清晰边缘的“块”。所以，粒计算是研究信息分类，被分成的块是两两分离的划分还是两两可能有交的模糊分割；研究分成的粒度大小，不同粒度层之间的关系，粒度分解和合并等。

定义 1.4[64]　一个定制的近似空间为一个系统 $AS_{\#,S}=(U, I_{\#}, v_S)$，其中：

U 为对象的非空有限集；

$I_{\#}$：$U\to P(U)$，$P(U)$ 表示 U 的幂集，是一个不确定函数，不确定函数对每个对象 x 定义了一个相似对象的集合；

v_S：$P(U)\times P(U)\to[0,1]$是一个粗糙包含函数。

定义 1.5[64]　（信息粒）：称序对 IS=（U，A）为信息系统，其中 U 为论域，A 为有限属性集，$a\in A$，一个基本的信息粒定义为 $EF_B(x)$，这里 EF_B 为 $a=a(x)$的选择子的连接，$\|EF_B(x)\|_{IS}=\|\wedge_{a\in B} a=a(x)\|_{IS}$，其中 $B\subseteq A$，$x\in U$，$\|\bullet\|$为公式集 Φ 到幂集 $P(U)$ 的映射函数。

从本质上讲，“粒”即基本元素，信息粒是在基本集中具有相同或相似属性值的对象集合，一个基本粒相当于粗糙集的一个等价类，等价类也称为等价粒，如决策规则的前件、后件和规则本身等就是一种粒。

定义 1.6[64]　设信息系统 IS=（U，A），假设 S 为一粒序列，且 IS 中$\|\bullet\|_{IS}$ 的语义及其元素已被定义，扩展$\|\bullet\|_{IS}$ 到 S 上，定义

为$\|S\|_{IS}=\{\|g\|_{IS}: g\in S\}$。

定义 1.7[64]（**粒集**） 设信息粒集 G 以及 IS 中的 G 的粒 $\|\bullet\|_{IS}$ 已经被定义，扩展$\|\bullet\|_{IS}$ 到集合 $H\subseteq G$ 为$\|H\|_{IS}=\{\|g\|_{IS}: g\in \mathrm{H}\}$。

对于信息粒来说，有两个基本的度量：粒的包含度（Inclusion）和接近度（Closeness），下面对它们进行简要介绍。

两个信息粒 G，G'的包含度至少为 P 记为 $v_P(G, G')$，类似地，两个信息粒 G，G'的接近度至少为 P 记为 $cl_P(G, G')$。

在信息系统 IS=（U，A）中，粒由 $\mathrm{EF}_B(x)$定义，这里 EF_B 为 $a=a(x)$ 的选择子的连接，$\|\mathrm{EF}_B(x)\|_{IS}=\|\wedge_{a\in B} \quad a=a(x)\|_{IS}$，其中 $B\subseteq A$，$x\in U$，设 $G=\{\mathrm{EF}_B(x): B\subseteq A\&x\in U\}$，设$\alpha$、$\beta\in G=\{\mathrm{EF}_B(x): B\subseteq A\&x\in U\}$，$\alpha$，$\beta$之间的精确包含的定义为：$\|\alpha\|_{IS}\subseteq\|\beta\|_{IS}$，这里 $\|\alpha\|_{IS}$，$\|\beta\|_{IS}$ 分别为满足α，β的对象的集合，而非精确描述，如知识发现的关联规则里，可以通过两个阈值 t 与 t'来定义：

$\mathrm{support}_{IS}(\alpha、\beta)=card(\|\alpha\wedge\beta\|_{IS})\geqslant t$ 且

$\mathrm{accuracy}_{IS}(\alpha、\beta)=\dfrac{\mathrm{support}_{IS}(\alpha,\beta)}{card(\|\alpha\|_{IS})}\ldots t'$。

在一个给定的信息系统中，基本的信息粒包含可以按照不同的方式进行定义，例如：

$v^{IS}_{t,t'}(\alpha,\beta)$ 成立当且仅当 $\mathrm{support}_{IS}(\alpha,\beta)\geqslant t$ & $\mathrm{accuracy}_{IS}(\alpha,\beta)\geqslant t'$。

基本粒的接近度可以定义为：

$cl^{IS}_{t,t'}(\alpha,\beta)$ 成立，当且仅当 $v^{IS}_{t,t'}(\alpha,\beta)$ 成立且 $v^{IS}_{t,t'}(\alpha,\beta)$ 成立。

1.2.2 信息粒化

粒化是人类理解认识问题的一个内在的自然活动，信息的粒化相当于把原始复杂的问题分解为多个可定义的子问题，这样可以降低原始问题的计算复杂性。粒化问题随处可见，它是许多学科的共同研究课题，如：可以把时间信息粒化形成时间信息粒——

可定义的时间间隔。

在很多计算机系统模型中，往往把内存资源粒化为内存页的概念来做为它的基本操作单位。

尽可能地粒化数字图像为基本单元——个体像素，这样便于对一些大的实体进行处理和分析。

在处理问题时，往往趋向于构造规则（if-then 描述），这其实已经将问题粒化。

在一个确定的程度上进行数据压缩也是信息粒化的一个范例。

对于信息的粒化，有两点需要说明：

（1）信息的粒化是一个自然分层的结构，可以把时间粒化为年、月、日、小时和分钟这样一个分层结构。

（2）不同的理论进行粒化的方法是不同的：

集合论使用间隔（区间）分析进行粒化，它的基本粒是定义在实轴上的间隔（Intervals），其形式化的模型为：集合 A 定义为给定的论域 U（实数 R 的子集）到{0，1}上的二值映射：A：$U\rightarrow$[0，1]，这里 A（u）为 A 的特征函数。

模糊集为集合概念的泛化，模糊集基于元素对于模糊集合的隶属程度。模糊集通过它们的隶属函数来形式化地描述：A：$U\rightarrow$[0，1]，其隶属值越大，给定的元素 x 对于模糊集的附属程度越强。

粗糙集是给定实验数据中近似概念的合成。粗糙集中的基本粒是按照等价关系定义的等价类。给定定义在 $U\times U$ 上的不可分辨关系 R，则一个粗糙集由下近似和上近似构成，从根本上说，上、下近似的差别越大，则边界越明显，粗糙度越大。

1.2.3　粒计算概念

信息粒度与信息粒是不同且相关的两个概念。

文献[88，89]使用一个三元组（U，F，Γ）来描述一个问题，其中：

U 表示问题的论域，也就是要考虑的基本元素的集合；设 F 为属性函数，定义为 F：$U \rightarrow Y$，Y 表示基本元素的属性集合；Γ 表示论域的结构，定义为论域上各个基本元素之间的关系。

从一个较粗的角度看问题，实际上是对 U 进行简化，把性质相近的元素看成是等价的，把它们归为一类，整体上做为一个新元素，这样就形成一个粒度较大的论域[U]，从而把原始问题（U，F，Γ）转化为新层次上的问题（[U]，[F]，[Γ]）。

粒计算是一种新的软计算方法，它的基本成分是论域的子集、类和簇。在粒化计算中，主要涉及的问题有粒的描述、粒之间的关系和粒的计算等，它们可以用于很多领域，如聚类分析、概念的形式化、机器学习和数据挖掘等。

目前粒计算的研究主要从粒的构造和粒的计算两个方面展开。粒的构造包括粒的形式化、表示和解释。粒的形式化与表示粒构造的算法描述；粒的解释一般指粒构造的语义，它表明了为什么两个对象能够归为同一个粒，进一步说，信息的粒化依赖于可用的知识。粒的计算则指在解决问题中粒的利用，它也是从语义解释及算法描述两方面入手：一方面，人们需要解释粒之间的不同关系，如相近性（Closeness）、依赖性（Dependency）、关联性（Association）、定义与解释粒的操作；另一方面，人们需要设计粒计算的方法与工具，如近似性（Approximation）、推理（Reasoning）和推论（Inference）[84]。

粒计算与数据挖掘的关系可以从概念的形式化与概念关系的确定两方面来说明。在概念的形式化研究中，一般认为概念由内涵与外延构成：概念的内涵由相关的属性或特征构成，它表明了对象应用的有效性，概念的外延指相应的对象集，外延中的对象具有相同的特征或属性，也就是说，外延构成了对象具体的实例。从粒化计算的观点看，每个粒可以作为概念的一个实例。一旦概念被构造和描述，人们便可用粒构造相应的计算方法。特别地，人们可以根据概念的内涵与外延研究概念之间的关系，比如

子概念（Sub-concept）、无关性（Disjoint）、重叠概念（Overlap Concept）和部分子概念（Partial Sub-Concept）等，这些关系可以按照规则或关联度量来方便地表示[84]。

粒计算的操作对象是信息粒，而信息粒按照粒度的不同展现出不同的层次，即一种分层结构。按照所处理的问题，通常把相同或相似“尺寸”（粒度）的粒划为一层，如果需要了解更多的细节，则把它们划为较小的粒，而这些粒则处于另外一层，这样，按照粒度的不同，信息粒呈现一种自然的分层结构，如上面偏序格结构一样。

1.2.4　粒计算的研究方法与方向

粒计算的形成和发展积累了多种思想、模型、技术及方法论，对粒计算的研究可以从不同观点着手。现有的粒计算研究可以概括为三个主要观点：结构化思维、结构化问题求解和结构化信息处理，它们相互关联，又自成体系，形成了粒计算特有的三角形[87]。

1．粒计算三角形

在粒计算三角形中，哲学思想偏重思维的结构化，方法论注重问题求解的结构化，计算模式强调信息处理的结构化。多层次粒结构把这三个观点紧密地结合在一起，形成了研究粒计算的三个观点，如图 1-1 所示。

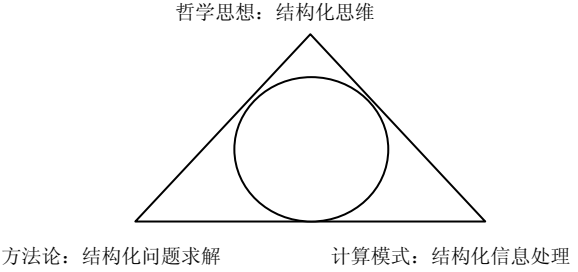

图 1-1　粒计算的三角形

结构化思维强调了对粒计算的哲学思想的研究。思维本身既是问题求解的必经过程和必要手段，也是人脑对信息进行筛选、分类比较、整理归纳的复杂处理。结构化思维是人类智能的重要体现形式。

结构化问题求解研究粒计算的方法论。张钹和张铃教授对这个问题有系统和详细的论述。从粒计算三角形的角度来看，抽象思维与哲学思想有关，抽象技能与方法论有关。问题求解是一个逻辑化和结构化的思维过程，它需要解决两个问题：一是构建问题的粒结构；二是在此粒结构中进行问题求解。在某些情况下，这两个任务的界限并不明显，需要放在一起综合考虑。

结构化信息处理注重以计算为主的问题求解。Bargiela 和 Pedrycz 在文献中对该问题有详细系统的描述。最近，Wing 提出了计算思维的观点，将结构化思维和结构化信息处理紧密联系在一起，她认为计算机科学家所采用的计算思维方式可以推广和应用到不同学科，这就需要不同抽象层次的思维。信息处理是问题求解的一个特例，它本身并不一定需要由计算机完成。信息处理至少可以分成抽象的、人脑中的和计算机中的信息处理三种形式。其中抽象的信息处理与粒计算哲学思想密切关联，构建对信息的粒结构分析；人脑中的信息处理遵循粒计算的普遍方法论，涉及诸如信息粒化的原则；计算机信息处理强调信息在计算机中的存储、计算、显示和传递[87]。

2．粒计算的三个层次

粒计算的三个研究观点之间还存在着层次关系。其中，结构化思维作为其哲学思想指导处于最高、最抽象的层次；结构化问题求解作为方法论处于中间层次；结构化问题求解可以进一步用结构化信息处理实现，因而结构化信息处理位于阶梯层次的底部，如图 1-2 所示。

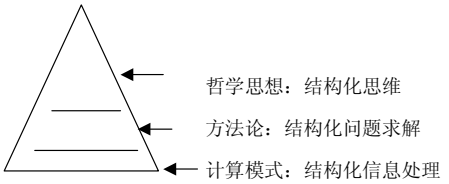

图 1-2　粒计算的三个层次

这样的层次结构反映了哲学思想、方法论及计算模式的关系。如果把粒计算研究作为一个实际问题，那么它也可以从以上三个层次求解。

对粒计算的研究方兴未艾，它的数学基础与理论还很不成熟，有许多问题还值得深入研究。

1.3　粗糙关系数据库

粗糙集理论经过近三十年的发展，基本理论的研究已日趋成熟，目前对粗糙集的研究主要集中在把粗糙集理论与相关学科的结合方面。粗糙集理论是从研究信息表（也称信息系统，或知识表达系统）的逻辑特性开始的，而关系数据库理论是从研究二维表开始的，信息表或信息系统实际上是数据库关系的泛化，这表明粗糙集理论和数据库理论有一种天然的联系，因此利用粗糙集理论和技术研究和解决与数据库有关的理论与应用问题是十分必要的。

在粗糙集与数据库关系的研究方面，已经获得很多研究成果。美国著名学者 T.Y.Lin 从信息表、知识依赖性、决策表与决策规则和部分依赖等方面研究了从关系数据库的观点看待粗糙集的问题[40]；J.W.Guan 与 D.A.Bell 从信息系统与数据库的关系、属性的分类、函数依赖、恒等依赖与关键字、寻找数据库所有的关键字、重要性与核、属性的重要子集和寻找关键字等方面研究

了信息系统的粗糙计算问题[41]；Beaubouef. Theresa Ann 博士通过对粗糙集理论和关系数据库理论的研究，提出把粗糙集与关系数据库相结合形成粗糙关系数据库（Rough Relational Database，RRDB），并以此为基础对粗糙关系操作算子、粗糙关系数据库的不确定性度量、粗糙函数依赖、精确数据的粗糙数据查询（RQCD）、模糊关系数据库模型、函数依赖与知识发现等专题进行了初步的研究，并把它们应用于地理信息系统中[42~43]；Shoji Hirano，Shusaku Tsumoto 等学者提出了利用粗糙集原理来进行数据库聚类分析的思想。近年来随着信息颗粒与粒化计算的出现，有许多学者开始把它们用于数据挖掘与知识发现，T.Y.Lin 发表系列论文研究了与粒计算有关的关系数据库的面向机器的数据挖掘建模理论问题。

Theresa Beaubouef Ann 在研究 Rough 集的基础上把粗糙集与数据库结合起来提出了粗糙关系数据库模型，并定义了各种粗糙关系操作算子。粗糙关系数据库模型（Rough Relational Database Model，RRDM）同普通的关系模型一样均是由包含元组的关系构成，元组 t_i 采用（d_{i1}，d_{i2}，…，d_{im}）的形式，其中 d_{ij} 是元组的一个属性值，隶属属性域 D_j。在普通的关系数据库模型中，$d_{ij}\in D_j$，在 RRDM 中，$d_{ij}\subseteq D_j$，$d_{ij}\neq\phi$，用 $P(D_i)$表示 D_i-ϕ 的幂集，D_i 为某一属性域。这里，RRDM 与普通的关系数据库的一个不同点是它的属性值可以由多个原子值构成，而不像关系数据库是单属性值[42]。RRDM 将粗糙集的重要性质引入到基本关系模型中，从而使之具有更好的检索能力和适应性[42~43]。在其博士论文中，重点探讨了粗糙关系数据库模型、模糊粗糙关系数据库模型、精确数据的粗糙查询和函数依赖和知识发现等专题；在随后的研究中，Theresa Beaubouef 对粗糙关系数据库的度量进行了探讨[90]，并提出了 Intuitionistic Rough Sets[91~95]，把粗糙关系数据库与 Intuitionistic Rough Sets 应用地理信息系统中。

在粗糙关系数据库研究方面，国内学者于近年来逐步开展了

一些研究。胡可云、眭跃飞、陆玉昌、王驹和石纯一等学者在文献[96]中提出了一种多值粗糙集模型，对基于共同关系对粗糙关系数据库的理论方面做了研究。该文提出了一种基于共同对的上下近似方法，能够更好地近似粗糙概念。此外该文还着重讨论了在多值情况下决策规则生成的情况，所提出的方法同样适用于传统粗糙集中需要生成组合决策规则的情形。在该论文集的另外一篇论文“Rough Relational Database:the Basic Definitions”中，作者给出了粗糙关系数据库几种不同的上下近似定义，分析了面向属性约简或从粗糙关系出发的粗糙集数据库分析的规则生成熵，并讨论了它们的基本性质。

曹付元和梁吉业等在文献[97]中研究基于 SQL 语言的粗糙数据查询，该文认为，查询在数据库中十分重要，而 SQL 是检索数据的一种强有力的工具，该文提出了一种粗糙关系数据库的粗糙查询方法，它扩展了标准 SQL 语言，实例表明了这种方法具有较好的应用前景。

郭景峰、李莉和宫继兵等在文[98]中等以粗集理论为研究方法，针对粗糙关系数据库属性值非原子性的特点，从语义等价的角度改进了已有的粗糙关系数据库函数依赖定义，提出了其修正定义粗函数依赖，使之更客观地反映粗糙关系数据库中数据的语义联系，体现现实世界不确定性信息的粗糙性和不完备性，并且给出了判断粗函数依赖是否成立的算法，并用粗关系实例验证了粗函数依赖的优越性，探讨了基于粗函数依赖的推理规则。而在另外一篇论文[99]中，郭景峰、宫继兵、李莉和刘佳等对 Rough 关系数据库上查询事务处理进行了研究，该文基于 Rough 集合理论和 Rough 关系数据库模型 RRDM(the Rough Relational Database Model)，将 Rough 精确查询和 Rough 完全查询分别与等价合类和共同类结合起来进行讨论和研究，提出了解决信息管理系统中查询事务处理问题的方法。

张熠、张金城和黄兵等在文献[100]中提出了一种粗糙查询的

SQL 实现新方法，该文认为，多值信息系统是对信息系统的自然扩展，查询是其中的重要组成部分。SQL 是检索、查询数据的有力工具，粗糙集理论是处理模糊和不确定知识的有力工具。运用粗糙集理论扩充 SQL，该文提出了一种处理粗糙关系数据库形式的多值信息系统的查询方法。分析显示，这种方法可以有效地处理多值信息系统的查询。

王丹、吴孟达和刘银山等在文献[101]中研究了粗糙关系数据库空间结构及其粗糙集模型，该文认为，粗糙关系数据库模型从本质上来说就是多值信息系统，它继承和扩展了经典的粗糙集理论，但与经典粗糙集理论又有着很大的不同。该文扩展了多值信息系统模型的定义，分析了粗糙关系数据库模型的空间结构，为粗糙数据库的不确定性进行度量提供一个依据，同时通过对粗糙关系数据库空间结构的分析，导出了基于粗糙关系数据库的上、下近似以及值约简，进一步基于粗糙关系数据库的空间结构定义了属性集之间的部分、传递依赖关系和粗糙关系数据库的规范，定义粗糙关系数据库的连接算子，并以此为基础得出了粗糙关系数据库的关系模式分解定理，为粗糙关系数据库模型的进一步研究提供了理论基础。

魏玲玲、邱桃荣和刘萍等在文献[102]中探讨了粗关系数据库中的数据更新。该文根据粗关系数据库中数据的特性，借助邻接表、十字链表存储不确定性数据，其中邻接表用于等价类的存储，十字链表用于数据库中基本表的存储。与传统的关系数据库更新不同，在粗关系数据库中更新基本表时，相应地等价类也要随之更新，该存储结构加快了对数据库中的数据更新速度。该文将算法与实例相结合，根据用户条件详细地讨论对等价类和粗糙关系数据库中基本表的数据更新。

邱卫根和徐相林在文献[103]中对基于粗糙集理论的粗关系数据库的熵进行了研究。熵是度量信息不确定性的重要工具，粗集数据分析方法是研究粗糙关系数据库熵的重要方法，该文首次利

用复合粗近似算子概念和方法，由属性值域上的二元关系导出了粗关系模式实例元组之间的二元关系，为利用粗集理论来研究粗关系数据库提供了必须的前提条件。在此基础上，提出了基于粗集的粗关系模式及其实例的信息熵和粗糙熵的概念，同时给出了它们的计算公式，最后以一个工程实例的计算验证了所提方法的有效性。

邱桃荣、葛寒娟、魏玲玲、徐苏和姚晓昆等在文献[104]中对基于相似度的粗关系数据库的近似查询进行了研究。该文基于数据库理论和粗集方法，研究了粗关系数据库中不确定数据的存储、索引和检索，提出了分别采用邻接表和十字链表实现粗糙关系数据库中属性值等价类和元组数据的存储；借助汉明距离和聚类方法，提出了实现粗糙关系数据库索引的方法；给出了一种基于 noash 集中的上、下近似计算数据间的相似度，并基于相似度给出了对粗糙关系数据库进行查询的模型，设计了相应的查询算法。最后，通过一个具体实例说明了查询算法的可行性和有效性。

马垣教授在专著《非经典关系数据库中》专门设置一章（第七章）研究粗糙关系数据库[105]，在该章中，共分“粗糙集的基本概念”、“粗糙关系数据库”、“粗糙关系查询”、“粗糙关系运算”、“粗糙运算符的性质”和“粗糙关系中的信息熵”几个专题。值得一提的是，该书对于 Theresa　Beaubouef　Ann 所给出的粗糙操作算子进行了大量的补充和扩充，使得粗糙关系数据库的粗糙操作算子更加容易理解，并且在此基础上对于所涉及的几个专题都给出了翔实易懂的例子，最后该书详细地介绍了粗糙关系中的信息熵。

作者在攻读博士学位及从事博士后研究工作期间，对粗糙集与关系数据库的关系、粗糙关系数据库模型、粗糙关系数据库与模糊关系数据库的关系、粗糙数据查询、粗糙函数依赖及其推理机制、基于粗糙集与信息颗粒的聚类方法、信息系统函数依赖的信息颗粒原理与计算、基于粗糙集的关系数据库范式及粗糙函数

依赖的近似度量等专题进行了初步研究，取得了部分成果[106~123]。

综上所述，目前国内外学者的研究主要聚焦在对粗糙关系操作算子、利用信息论研究粗糙关系数据库的不确定性度量、粗糙函数依赖、粗糙数据查询、粗糙关系数据库的分解算法、粗糙关系数据库的存储结构、粗糙范式理论及粗糙关系数据库的知识发现等专题，并把它们应用于地理信息系统、医学诊断及农业分析中等。

第 2 章　粗糙集与RDB关系研究及RRDM

是故大知观于远近，故小而不寡
——《庄子·秋水篇》

2.1　引言

粗糙集数据分析[41]技术（Rough Set Data Analysis，RSDA）主要用来分析数据的特性、数据库中属性的依赖性、属性的重要性及决策规则抽取等。RSDA 是一种条件数据分析技术，它依赖于属性和度量模型的选择。粗糙集理论中的知识表达方式一般采用信息表或信息系统。

本章首先以粗糙集数据分析技术为工具，对关系数据库（Relational Database，RDB）理论和粗糙集理论的关系进行系统研究。具体地，从粗糙集理论与 RDB 理论产生的背景、数据库关系与信息表的形式化语义比较和两种理论核心概念之间的关系等方面对它们的关系进行了深入和全面的探讨。然后以 RRDB 为基础结合 Pawlak 代数对粗糙关系数据库模型及其关系操作进行了分析与完善，在此基础上提出了粗糙分解算子的概念，最后对粗糙关系数据库与模糊关系数据库的关系进行了初步的研究。

2.2 RDB理论与粗糙集理论关系的研究

2.2.1 RDB与粗糙集产生的背景比较

1970 年，IBM 公司的研究员 E.F.Codd 发表了题为“大型共享系统的关系数据库的关系模型”的论文，首次提出了数据库系统关系模型的概念。20 世纪 80 年代以来，各大计算机厂商新推出的数据库管理系统几乎都支持关系模型，可以说关系模型是目前最重要的一种数据库系统模型，也是最有发展前途的一种数据模型。

关系模型具有以下特点：首先，它是建立在严格的数学理论基础上的，运用数学的方法来研究数据库的关系运算，因此具有严密的数学基础；其次，关系模型概念单一，无论实体还是实体之间的联系都用关系来表示，操作的对象和结果均为关系；再次，关系模型的存取路径对用户透明，从而具有更高的数据独立性和更好的安全保密性；最后，关系必须是规范化的关系，即每一个关系必须满足一定的要求或称为规范条件[124~125]。

关系模型由数据结构、关系操作集合和关系的完整性三个部分组成。关系模型的数据结构为关系（二维表结构）；关系操作集合指关系模型提供的一组完备的关系运算，以支持对数据库的各种操作，关系运算分为关系代数和关系演算两部分，其中又以关系代数尤为重要；关系的完整性指关系模型的三类完整性规则。

20 世纪 70 年代，波兰学者 Z.Pawlak 和波兰科学院、波兰华沙大学的一些逻辑学家们一起从事关于信息系统逻辑特性的研究，他们针对从实验中得到的以数据形式表述的不精确、不确定和不完整的信息和知识，进行了分类分析，这一研究成为粗糙集理论产生的基础。1982 年，Z.Pawlak 发表经典论文 Rough Sets，

标志着粗糙集理论的诞生。

可以看出，RDB 理论和粗糙集理论都是从研究二维表出发的（信息表也是二维表），但由于它们产生的原因不同，因此具有不同的哲理，能够解决的问题也不同，所以它们向不同的方向发展。RDB 理论认为数据的语义是已知的，并且通过数据的语义来组织数据；粗糙集理论认为数据的语义可由给定的数据来定义，并且通过这些可用的数据来发现新的模式、规则和数据语义[40]。从哲学意义上看，粗糙集理论是解决模糊性和不确定性问题的一种新方法，它是模糊方法的补充和完善；从数据处理意义上说，粗糙集理论是进行数据分析的一种新技术，粗糙集数据分析技术是数据库数据处理技术的补充，是对数据库数据处理技术的新发展，粗糙集理论是一种新的数据挖掘理论。

2.2.2　关系与信息表的形式化语义比较

在 RDB 中，关系模式是一个属性的有限集 $T=\{A_1, A_2, \cdots, A_n\}$，每个属性 A_i (i=1, 2, …, n)对应于一个值域 D_i，关系模式上的一个关系 R 是 $D_1\times D_2\times\cdots\times D_n$ 的一个子集，关系实质上是一张二维表，它的每一行称为一个元组，或关系的一个记录，一个关系通常是由赋予它的元组语义来确定的，元组语义实质上是一个 n 元谓词（n 为属性的个数），因此又可以把关系描述为二元组$<U, T>$, $U=\{r_1, r_2, \cdots, r_m\}$为元组的有限集，$T=\{A_1, A_2, \cdots, A_n\}$是属性的有限集，Dom($A_i$)是属性 A_i 的值域，Dom(T)=Dom(A_1)∪Dom(A_2)∪…∪Dom(A_n)，其中每个元组 t 可以由下列映射表示：t：$T\rightarrow$Dom(T)，这里 $A_i\in T$，$t(A_i)\in$Dom(A_i)。

粗糙集理论的知识表达方式为信息表（又称信息系统，知识表达系统），它同样可以表示成二元组$<U, T>$, $U=\{u_1, u_2, \cdots, u_m\}$为对象的有限集，称为论域或对象空间，$T=\{A_1, A_2, \cdots, A_n\}$是属性的有限集，Dom($A_i$)是属性 A_i 的值域，Dom(T)=Dom(A_1) ∪ Dom(A_2) ∪ … ∪ Dom(A_n))，其中每个元组 t 可以由下列映射表示：

ρ：$U\times T\rightarrow \mathrm{Dom}(T)$，它是一个信息函数，$\rho(u, A_i)\in \mathrm{Dom}(A_i)$，这样每个元组可以表示成 $t=(\rho(u, A_1), \rho(u, A_2), \cdots, \rho(u, A_n))$[40]。

从以上的形式化描述中，可以看出 RDB 与粗糙集在知识表达上是有所区别的，主要表现在：

（1）规范化的差别。关系的行是元组的集合，而信息表把每一行看作是一个对象的信息，关系的任意两行是不能相同的，这是由于关系得满足规范化的要求，而信息表在关系的基础之上增加了对象罗列，而且不同的对象可以有相同的知识表达（即有相同的行），它不受关系数据库范式的约束，可以说，信息系统或信息表是 RDB 关系的泛化形式。

（2）映射上的差别。关系把属性映射到相应的值域上，而信息表把对象及对应的属性映射到相应的值域上，信息表的映射函数一般要涉及具体的对象信息。

（3）语义上的差别。在 RDB 中，关系即二维表不再分类，而信息表则不同，它可分为有决策属性信息表和无决策属性信息表。在有决策属性信息表中，属性集 $T=C\cup D$，其中 C 为条件属性，D 为决策属性，这时的信息表便称为决策表，或决策信息系统，决策表是一种特殊的信息表，此时决策表的一行可以解释为一个决策规则，即 if $t(X)=c$ then $t(Y)=d$；而关系中的一行只是一个简单的记录。

综上所述，RDB 中的二维表与粗糙集中的信息表是既有联系又有区别的，当信息表中没有重复行，并消除对象列标识的影响时，可以认为信息表即是 RDB 中的关系。

2.2.3 两种理论核心概念之间的关系研究

粗糙集最基本的几个概念是知识的分类和知识库、等价关系、依赖性、约简和核。RDB 中的重要概念有关系、函数依赖和键（关键字）。下面来研究这些概念之间的关系。

1）关系属性与信息表等价关系

粗糙集是建立在分类机制的基础上的，分类是粗糙集数据分析的基本技术，它将分类理解为特定空间上的等价关系，而等价关系构成了对该空间的划分，论域 U 的一个划分称为 U 的一个知识库。任何子集 $X \subseteq U$，称为 U 的一个概念，U 中的任何概念族称为关于 U 的抽象知识，简称知识。U 的一个划分定义为：$\{X_1, X_2, \cdots, X_n\}$，$X_i \subseteq U$，$X_i \cap X_j = \phi$；对于 $i \neq j$，$i, j=1, 2, \cdots, n$；$\bigcup_{i=1}^{n} X_i = U$[5]。

设 R 是 U 上的一个等价关系，U/R 表示 R 的所有等价类构成的集合，$[x]_R$ 表示包含元素 x 的等价类，R 是 U 上的等价关系。一个知识库可以用 $K=(U, R)$ 表示。若 $P \subseteq R$，且 $P \neq \phi$，则 $\cap P$（P 中所有等价关系的交集）也是一个等价关系，称为 P 上的不可分辨关系，记为 $\mathrm{ind}(P)$。不可分辨关系即等价关系，对于一个属性子集 $B \subseteq T$，有 $\mathrm{IND}(B)=\cap\{\mathrm{IND}(A_i)：A_i \in B\}$，可以得出如下结论：

（1）表中的每个属性是一个等价关系。

（2）关系表中的属性子集是一个等价关系。

通过这两点结论，可以认为粗糙集是从不同的角度来研究 RDB 的关系的，它把关系看作是一个信息表，从研究知识的逻辑特性出发，利用分类机制来研究关系：把它的每行看作是对某个对象的描述，把它的一列或若干列看作一个等价关系。

2）粗糙集与 RDB 的依赖性比较

定义 2.1[126]　（属性依赖性及度量）设 $S=<U, A>$ 为信息系统，P，$Q \subseteq A$ 为属性子集，属性集 Q 依赖于属性集 P 的程度为：

$$K=\gamma_P(Q)=\frac{card(POS_P(Q))}{card(u)} \qquad (2.1)$$

其中 $POS_p(Q)$ 为 Q 的 P 正域。

当 $K=1$ 时，S 中属性集 Q 全依赖于属性集 P，当 $0<K<1$

时，S 中属性集 Q 部分依赖于属性集 P，当 K=0 时，属性集 Q 独立于属性集 P。

定义 2.2[126]（知识的依赖性及度量）设 $K=(U, R)$为一知识库，P，$Q \subseteq R$，有：

（1）知识 Q 依赖于知识 P，当且仅当 $\mathrm{IND}(P) \subseteq \mathrm{IND}(Q)$；

（2）知识 Q 等价于知识 P，当且仅当 $P \Rightarrow Q$ 且 $Q \Rightarrow P$，记为 $P \equiv Q$；

（3）知识 Q 与知识 P 是独立的，当且仅当 $P \Rightarrow Q$，$Q \Rightarrow P$ 均不成立。

显然，$P \equiv Q$，当且仅当 $\mathrm{IND}(P)=\mathrm{IND}(Q)$。

知识之间依赖性的解释也与属性依赖性的解释类似。

定义 2.3（数据库函数依赖）设 $R=\{A_1, A_2, \cdots, A_n\}$为具有 n 个属性的关系模式，X，$Y \subseteq R$，X，Y 之间的函数依赖记为 $X \rightarrow Y$，它意味着为当 $t_i[X]=t_j[X]$时，必有 $t_i[Y]=t_j[Y]$，其中 t_i，t_j 为元组，称 X 函数决定 Y，或 Y 函数依赖于 X。

信息系统的函数依赖与数据库的函数依赖类似，J.W.GuAn 与 D.A.Bell 在文[41]中已经证明了下面的定理：

引理 2.1 函数依赖 $X \rightarrow Y$ 可以表示为：$\theta_X \subseteq \theta_Y$，即 $\bigcap_{a \in X} \theta_a \subseteq \bigcap_{a \in Y} \theta_a$，这里是 θ_X，θ_Y 是 U 上的两个等价关系。

该定理表明了信息系统函数依赖与等价关系之间的关系，它也说明对于信息系统函数依赖的判断可以转化为对相应的等价关系的判断来完成；同时由于信息系统是泛化的关系数据库，而关系数据库是信息系统的特例，所以该定理对于关系数据库也是适合的。

定义 2.4[41]（恒等依赖）信息系统 $S=<U, A>$ 中，任意两个属性子集 X，$Y \subseteq A$ 之间的恒等依赖描述为：$X \leftrightarrow Y$，它在信息系统中成立，当且仅当 $X \rightarrow Y$ 且 $Y \rightarrow X$。

按照上述定理，恒等依赖 $X \leftrightarrow Y$ 可以描述 $\theta_X = \theta_Y$，即 $\bigcap_{a \in X} \theta_a = \bigcap_{a \in Y} \theta_a$。

分析粗糙集的依赖性与数据库的依赖性，可以得出下列结论：

（1）从概念上讲，经典的关系数据库理论中的数据依赖一般研究函数依赖、多值依赖、连接依赖和投影依赖等，而粗糙集对于依赖性的研究涉及关系数据库中的函数依赖，但不涉及多值依赖和连接依赖；除此之外，粗糙集增加了知识依赖性和属性的恒等依赖性的研究。属性的依赖性是从列上研究属性集之间的关系，而知识依赖则是从行上利用等价类来研究知识的；知识依赖与恒等依赖从概念上拓宽了数据库依赖性的研究领域。

（2）从判定方法上看，RDB 通常是根据数据库中元组的语义来确定相应的函数依赖，然后利用范式理论对它们进行规范化，而粗糙集的数据依赖和知识依赖不但涉及语义方面，而且使依赖性的判定可以通过具体的数学方法来完成，式（2.1）说明 RDB 的函数依赖相当于粗糙集的完全依赖（K=1）的情况。

（3）虽然关系数据库中也有部分依赖的概念，但是它与粗糙集中的部分依赖是有区别的。在 RDB 中，对于关系模式 R（U），如果 $X \to Y$，且对于 X 的任何一个真子集 X'都有 $X' \to Y$ 不成立，则称 Y 对 X 完全函数依赖，若 $X \to Y$，但 Y 不完全函数依赖于 X，则称 Y 对 X 部分函数依赖。从这个定义可以看出，RDB 的部分函数依赖只涉及属性集与其属性子集之间的依赖性研究，而粗糙集的部分依赖不仅包括属性集与其属性子集之间的依赖性研究，还包括无包含关系属性集之间的依赖性研究，另外粗糙集的部分依赖还包含了知识间的度量研究，并在此基础上给出了依赖性的度量方法，使得依赖性的度量定量化，把依赖性的研究又向前推动了一步。

3）粗糙集属性的重要性、约简、核、RDB 的键之间的关系

定义 2.5[41]　对于一信息系统 S=<U，A>，A 为它的属性集，$X \subseteq A$，属性集 X 的子集 X_0 称为 X 的一个键（关键字），则 X_0 应满足：

（1）恒等依赖：$\theta_{X_0}=\theta_X$，即 $X_0 \leftrightarrow X$；

（2）最小化：如果 $Y \subset X_0$，那么 $\theta_X \subset \theta_Y$，即 $Y \leftrightarrow X$ 不成立。

这是很自然的，因为 X_0 为键，它可以唯一地确定一个元组，因此 $ind(X_0)=ind(X)$，即 $X_0 \leftrightarrow X$，而键的真子集不再是键，因此，它满足最小化要求。

定义 2.6[41] （属性的重要性）设 $S=<U, A>$为信息系统，对于任意的 $X \subseteq A$，设 $x \in X$ 当 $\theta_X \subset \theta_{X-\{x\}}$，属性 x 是重要的；当 $\theta_X = \theta_{X-\{x\}}$，属性 x 是不重要的。

定义 2.7[42] （属性的重要性度量）

设 $S=<U, A>$为信息系统，对于任意的 $X \subseteq A$，$x \in X$，x 的重要性定义为：

$$sig_{X-\{x\}}(x)=\frac{|U/\theta_X|-|U/\theta_{X-\{x\}}|}{|U|}$$

显然当 $sig_{X-\{x\}}(x)>0$ 时，属性 x 是重要的。

定义 2.8[41] 设 X 是 A 的非空子集，$\phi \subset X \subset A$，属性子集 X 是重要的，当且仅当 $x \in X$ 中每个 x 是重要的，否则是不重要的。

定义 2.9[41] （属性集 X 的核）设 $S=<U, A>$为信息系统，设 X 是 A 的非空子集，$\phi \subset X \subset A$，对于任意的 $x \in X$，X 中所有重要属性的集合称为 X 的核，记为：

$$C_X=\{x \in X | sig_{X-\{x\}}(x)>0\}$$

通过对于属性重要性及其度量、约简及键、核的研究，可以得出以下结论：

（1）比较数据库的键与粗糙集的属性约简，可以看出键与约简都满足恒等依赖与最小化要求，因此可以认为 RDB 中的一个键是粗糙集的一种特殊形式的约简，而粗糙集的约简不一定是 RDB 的一个键，因为在很多情况下它不具有键的函数决定的特性，当然，键与约简一般并不唯一。

（2）核是所有重要属性的集合，也是所有键的交集。每个键中的属性并不一定都是重要属性，但每个键中一定包含重要属性，当数据库只有一个键时，它也是信息表的核。

（3）某些属性子集是重要的，但它未必是数据库的键。

（4）某些属性子集是不重要的，它一定不是数据库的键。

（5）当属性子集是唯一的键时，这时它既是重要的，同时也是信息表的核。

在分析粗糙集与关系数据库的关系时，注意到粗糙集对 RDB 的一个特殊贡献也是 RDB 所不具备的功能——粗糙度量。这些度量包括粗糙集的近似精度、近似分类质量、近似分类进度和属性的依赖性度量、属性的重要性度量和规则的可信度、知识的依赖性度量等。梁吉业教授已经证明了这些度量均可归结为包含度[6]。

关系数据库本身并没有对信息不确定性的度量，有了粗糙度量，就可以定量分析知识粒度的情况，分析数据库中的属性重要性情况及知识库之间的依赖程度，分析规则的可信度等，这使得对数据库中的不确定性问题有了一个量化的描述。

Theresa　Beaubouef　Ann 在研究粗糙集的基础上把粗糙集与数据库结合起来提出了粗糙关系数据库模型（简称 RRDM），并定义了各种粗糙关系操作算子。RRDM 将粗糙集的重要性质引入到基本关系模型中，从而使之具有更好的检索能力和适应性[42~43]。具体地说，该模型把粗糙集的不可分辨关系、等价类、上近似、下近似等概念与 RDB 的基本理论相结合，改进了 RDB 的检索能力，使得查询结果返回的是一个粗糙关系，即其结果不仅包含一个确定的应答（下近似），而且包含了可能的应答（上近似），在 RRDM 的研究中，Theresa　Beaubouef 还通过把等价关系一般化为相似（相容）关系创造性地提出了粗糙函数依赖的概念，丰富了函数依赖的研究领域，当然粗糙函数依赖并不是粗糙集本身的概念，它是粗糙集与 RDB 结合的产物。RRDB 模型的检索能力也可以给予启示：等价类的选择实际与关系数据库的“选择”操

作有着相同的内涵[113~114]。

2.3 对RRDM的研究

2.3.1 引言

定义 2.10 粗糙关系数据库模型（Rough Relational Database Model，RRDM）同普通的关系模型一样均是由包含元组的关系构成，元组 t_i 采用（d_{i1}，d_{i2}，…，d_{im}）的形式，其中 d_{ij} 是元组的一个属性值，隶属属性域 D_j。在普通的关系数据库模型中，$d_{ij}\in D_j$，在 RRDM 中，$d_{ij}\subseteq D_j$，$d_{ij}\neq\phi$，用 $P(D_i)$表示 $D_i-\phi$ 的幂集，D_i 为某一属性域。这里，RRDM 与普通的关系数据库的一个不同点是它的属性值可以由多个原子值构成，而不像关系数据库是单属性值[42]。

定义 2.11 粗糙关系 R 是粗糙元组的有穷集合，它是集合叉积 $P(D_1)\times P(D_2)\times\cdots\times P(D_m)$的一个子集。

在关系数据库中，关系 R 的值用 r 或 $r(R)$表示，它是 n 个元组的集合，$r=\{t_1, t_2, \cdots, t_n\}$，每个元组 t_i 可以表示为（v_{i1}，v_{i2}，…，v_{im}），$v_{ij}\in D_j$，$1\leqslant i\leqslant n$，即 $t_i\in D_1\times D_2\times\cdots\times D_m$，而 $r\subseteq D_1\times D_2\times\cdots\times D_m$。

在粗糙关系数据库中，粗糙关系的定义是由关系数据库中的关系定义引申而来的，在 RRDM 中，它的一个元组可以称为粗糙元组，任意一个粗糙元组 t_i 是粗糙关系 R 的一个成员，这就意味着它也是 $P(D_1)\times P(D_2)\times\cdots\times P(D_m)$的一个成员，即 $t_i\in P(D_1)\times P(D_2)\times\cdots\times P(D_m)$。

定义 2.12 粗糙关系 R 是粗糙元组的有穷集合，它是集合叉积 $P(D_1)\times P(D_2)\times\cdots\times P(D_m)$的一个子集。

定义 2.13[42] 粗糙元组 $t_i=(d_{i1}, d_{i2}, \cdots, d_{im})$的一个解释 $\alpha=(A_1, A_2, \cdots, A_m)$：它意味着对$\forall j(1\leqslant j\leqslant k)$来说，进行值的分

配，即使 $A_j \in d_{ij}$。

传统的关系数据库由于其属性值是原子的，所以其解释即元组自身，而粗糙关系数据库，其属性值可能是由多个原子值组合而成的属性值，因此便有了对它的多个指派，即对其每个 d_{ij} 抽取其一原子值而形成该元组的一个指派；即解释。

定义 2.14　粗糙关系数据库模式

设 R 为一粗糙关系，它的属性为 A_1，A_2，…，A_m，其对应的域为 $P(D_1)$，$P(D_2)$，…，$P(D_m)$，则粗糙关系可以表示为：

$$R=(A_1/P(D_1)，A_2/P(D_2)，\cdots，A_m/P(D_m))$$

或 $R=(A_1，A_2，\cdots，A_m)$。

其中 A_i 也称为粗糙关系 R 的等价类，或等价关系，它的全体构成了一个等价类簇，上式是对粗糙关系的类型的描述，称为粗糙关系 R 的模式。

定义 2.15　对于粗糙关系数据库 $R=(A_1，A_2，\cdots，A_m)$来说，若其中某一属性或属性组的值（原子的或非原子的）能够唯一地决定一个元组，而其他任何真子集无此性质，则该属性或属性子集称为该粗糙关系数据库的键（记做 Rough-KEY（R））。

2.3.2　Rough关系操作算子及其性质

关系代数（Relational Algebra）是在集合代数基础上发展起来的，其数据的操作可分为传统的集合运算和专门的关系运算两类。传统的集合运算包括并（Union）、差（Difference）、交（Intersection）和笛卡儿积（Cartesion Product），专门的关系运算包括选择（Select）、投影（Project）、连接（Join）和除（Division）。

关系代数中五个基本的操作并（Union）、差（Difference）、笛卡儿积（Cartesion Product）、选择（Select）和投影（Project）组成了关系代数完备的操作集。

一般地，把由 RS=（2^U，$\cup$，$\cap$，$\backsim$，$\underline{apr}$，$\overline{apr}$）所构成

的代数系统称为 Pawlak 粗集代数，它是标准集合代数（2^U，$\cup$，$\cap$，$\sim$）的扩展，Pawlak 粗集代数便是以它为基础形成的，Rough 操作符算子集为（$\sim$，$\cup$，$\cap$，σ，Π，∞）[19]。

下面对文献[42]中的六种粗糙操作进行分析。

定义 2.16 （并兼容）两个粗糙关系 X（A_1，A_2，…，A_m），Y（B_1，B_2，…，B_m），如果在它们的关系模式中有相同的属性数，并且对所有的 i=1，2，…，m，A_i 与 B_i 的域相同，则称它们为并兼容。

1）粗糙差算子

定义 2.17 对于两个并兼容的粗糙关系 X，Y 来说，$X-Y$ 的结果是一个粗糙关系 T，它满足：

$$\overline{R}\,T=\{t|t\in \overline{R}\,X\wedge t\notin \overline{R}\,Y\}$$
$$\underline{R}\,T=\{t|t\in \underline{R}\,X\wedge t\notin \underline{R}\,Y\}$$

即 T 的下近似是 X 的下近似与 Y 的下近似的差，T 的上近似是 X 的上近似与 Y 的上近似的差。

2）粗糙并算子

定义 2.18 对于两个并兼容的粗糙关系 X，Y 来说，$X\cup Y$ 的结果是一个粗糙关系 T，它满足：

$$\overline{R}\,T=\{t|t\in \overline{R}\,X\cup \overline{R}\,Y\}$$
$$\underline{R}\,T=\{t|t\in \underline{R}\,X\cup \underline{R}\,Y\}$$

即 T 的下近似是 X 的下近似与 Y 的下近似的并，T 的上近似是 X 的上近似与 Y 的上近似的并。

3）粗糙交算子

定义 2.19 对于两个并兼容的粗糙关系 X，Y 来说，$X\cap Y$ 的结果是一个粗糙关系 T，它满足：

$$\overline{R}\,T=\{t|t\in \overline{R}\,X\cap \overline{R}\,Y\}$$
$$\underline{R}\,T=\{t|t\in \underline{R}\,X\cap \underline{R}\,Y\}$$

即 T 的下近似是 X 的下近似与 Y 的下近似的交，T 的上近似是 X 的上近似与 Y 的上近似的交。

4）粗糙选择算子

关于粗糙选择算子，Theresa　Beaubouef　Ann 给出的原始定义为：

定义 2.20　粗糙选择$\sigma_{A=a}(X)$是从 X 中选择的元组，其结果是一个粗糙关系 Y，它具有与 X 相同的模式，设 $A=\{a_i\}$，a_i，$b_j \in \mathrm{dom}(A)$，则：

$$\overline{R}\ Y=\{t|t\in X \wedge \cup_i[a_i]\subseteq \cup_j[b_j],\ a_i\in a,\ b_j\in t(A)\ \}$$

$$\underline{R}\ Y=\{t|t\in X \wedge \cup_i[a_i]=\cup_j[b_j],\ a_i\in a,\ b_j\in t(A)\ \}$$

这里选择结果 Y 的上、下近似也是按照粗糙集来定义的。

在文献[105]中，马垣教授对粗糙选择算子给出了一个更为简洁明了的定义：

设 R 是一个关系模式，r 是 R 上的粗糙关系，$A\in R$，$a\subseteq \mathrm{dom}(A)$，则对 r 的粗糙选择：$\sigma_{A=a}(r)$,是 R 上的一个粗糙关系 s, s 的上近似及下近似为：

$$R^*(s)=\{t\in r|[t(A)]r_A\supseteq[a]\ r_A\ \}$$

$$R_*(s)=\{t\in r|[t(A)]\ r_A=[a]\ r_A\ \}$$

即 s 的上近似在 A 上取值的等价类是 a 的等价类的超集的那些元组，s 的下近似在 A 上取值的等价类等于 a 的等价类的那些元组。

与粗糙选择算子相关的性质如下：

设 X，Y 为具有模式 R 的关系，$\gamma\in\{\cup、\cap、-\}$为布尔集合意义下的算子，则

$\sigma_{A=a}(X\gamma Y)=\sigma_{A=a}(X)\gamma\sigma_{A=a}(Y)$。

5）粗糙投影算子

关于粗糙投影算子，Theresa　Beaubouef　Ann 给出的原始定义为：

定义 2.21　X 在 B 上的粗糙投影记为$\prod_B(X)$，是一个具有模式 $Y(B)$ 的关系 Y：$Y(B)=\{t(B)|t\in X\}$。

对于粗糙投影算子，马垣教授给出了一个扩充而清晰的定义：

设 R 是一个关系模式，r 是 R 上的一个粗糙关系，$X \subseteq R$,则 r 在 X 上的投影记做$\prod_x(r)$，是一个粗糙关系 s，s 的上近似及下近似为：

$$R^*(s)=red\{t(X)|t\in R^*(r)\}$$

$$R_*(s)=red\{t(X)|t\in R_*(r)\}$$

即 s 的上近似是 r 的上近似在 X 上的取值的集合删去冗余后的结果，s 的下近似是 r 的下近似在 X 上的取值的集合删去冗余后的结果。

这里，*red* 表示删去冗余，删去冗余的方法如下：

（1）若两个相同元组同时出现在$\{t(X)|t\in R^*(r)\}$或同时出现在$\{t(X)|t\in R_*(r)\}$中，则删掉其中任何一个均可；

（2）若两个相同元组一个出现在$\{t(X)|t\in R^*(r)\}$中，另一个出现在$\{t(X)|t\in R_*(r)\}$中，则删掉出现在$\{t(X)|t\in R^*(r)\}$中的那一个。

粗糙投影算子具有以下性质：

对于具有模式 R 的关系 Y，若 $X_1 \subseteq X_2 \subseteq X_3 \subseteq \cdots \subseteq X_n$ 成立，则只有最外层的算子是必要的，虽然如此，利用等价类值处理冗余元组的删除问题：

$$\Pi_{X_1}(\Pi_{X_2}(\cdots\Pi_{X_n}(Y)\cdots))=\Pi_{X_1}(Y)$$

6）粗糙连接算子

定义 2.22 （1）粗糙条件连接。设 X（A_1，A_2，…，A_m），Y（B_1，B_2，…，B_n）分别为具有 m 个和 n 个属性的粗糙关系，则：

$X_{\infty<\text{连接条件}>}Y$ 的结果为 T（C_1，C_2，…，C_{m+n}），这里 T 为：

$$\{t|\exists\, t_x\in X,\ t_Y\in Y,\ \ for\ \ t_x=t(A),\ t_Y=t(B)\},$$

并且 $t_x(A\cap B)=t_Y(A\cap B)$，for $t\in \underline{R}\,T$，

$t_x(A\cap B)\subseteq t_Y(A\cap B)$ 或 $t_Y(A\cap B)\subseteq t_x(A\cap B)$ $t\in \overline{R}\,T$，其

<连接条件>是一个或多个类似 $A=B$ 形式的条件。

（2）无条件粗糙连接。粗糙连接也可以有无条件连接，在这种情况下，连接则变成了笛卡尔积。

（3）粗糙自然连接。在上述条件连接的<连接条件>中，把这个条件表示为 $A_i \theta B_j$，如果这个条件中的 θ 为=，则此种连接可以称为粗糙对等连接。在对等连接中，如果对应属性的值是相同的，则出现列的冗余，没有必要重复列出，只要在两个对应属性中取一个就可以了，这种消除冗余属性的对等连接称为粗糙自然连接。

7）除算子

Theresa Beaubouef Ann 在其博士论文中还给出了普通的关系数据库所具有的除操作算子，定义如下：

设 $X(A)$与 $Y(B)$为具有关系模式 $B \subseteq A$ 的两个粗糙关系，设模式 $C=A-B$。X 与 Y 的商 $T=X \div Y$ 是一个粗糙关系 T（C）,这里：

$$\overline{R}\ T=\{t \mid \forall t_Y \in \overline{R}\ Y，\ \exists t_X \in \overline{R}\ X \wedge t \subseteq t_X(C) \wedge t_Y \subseteq t_X(B)\}$$

$$\underline{R}\ T=\{t \mid \forall t_{\underline{Y}} \in \underline{R}\ Y，\ \exists t_X \in \underline{R}\ X \wedge t=t_X(C) \wedge t_Y=t_X(B)\}$$

直观地说，$X \div Y$ 是 X 在模式 C 上使得 T 能够与 Y 连接而包含在 X 中的投影的最大粗糙子集。除操作算子可以利用关系操作序列（$\prod$，×，–）来等价地表示：

$$X \div Y=\prod_C(x)-\prod_C[(Y \times \prod_C(x))]。$$

2.3.3　粗糙分解算子

粗糙关系数据库的属性值具有非原子性，在某些时候，为了操作的需要，需要对数据库所有属性值只取其中的一部分，这样所形成的粗糙元组的集合即粗糙关系便构成了粗糙关系数据库的一个分解，并由此提出了分解算子[113]。

定义 2.23　给定粗糙关系 r_k 与 r_f，r_k 具有属性（A_1，A_2，…，A_m）它是 n 个元组的集合，其属性域为：（D_1，D_2，…，D_m），其属性值用 $r_k(A_{ij})$表示，粗糙关系 r_f 与 r_k 具有相

同的属性与属性域，它的属性值用 $r_f(A_{ij})$表示，对于任意的 i，j，若 $r_f(A_{ij}) \subseteq r_k(A_{ij})$，则称粗糙关系 r_f 为粗糙关系 r_k 的一个分解，记为 $r_f=\Gamma(r_k)$。

在特殊情况下，若粗糙关系 r 的每个元组均为 R 对应元组的唯一的一个解释，则此时 Rough 关系数据库 r 变为标准的关系数据库，也可以看出，标准的关系数据库是 Rough 关系数据库的特例。

定义 2.24 对于一个 RRDB 中所有的多值（复合属性值），取其任一属性值来代替该多值，其余单属性值保持不变，这样形成的关系为 RRDB 的一个单值分解。

若一 RRDB 的某一属性值为 $r(a_i)$，那么它的单值分解相同位置的属性值 $s(a_i) \in r(a_i)$。

定义 2.25 对于一 RRDB 中所有的多值，取其任一子集来代替该多值，其余单值保持不变，这样形成的粗糙关系为 RRDB 的一个多值分解。

若一个 RRDB 的某一属性值为 $r(a_i)$，那么它的多值分解相同位置的属性值 $s(a_i) \subseteq r(a_i)$。

在 RRDB 中数据查询中，需要用到上述的两种分解，用 Γ 表示分解算子[113]。

命题 2.1 设粗糙关系 r 的元组集合为（R_1，R_2，…，R_n），α_1，α_2，…，α_n 为 r 的 n 个解释，则由（α_1，α_2，…，α_n）形成的关系 s 必定为的 r 一个分解。

证明：根据解释的定义，该命题是显然的。

定理 2.1 若 s 为粗糙关系 r 的唯一分解，则必有 $\underline{R}\, r=s$，$\overline{R}\, r=r$。

证明：设 s 为粗糙关系 r 的唯一分解，r 由元组集合（t_1，t_2，…，t_n）组成，s 由元组集合（α_1，α_2，…，α_n）构成，k_{ij} 为 s 的任一元组 $\alpha_i(1 \leqslant i \leqslant n)$的任一属性值，$v_{ij}$ 为 r 的任一元组 t_i 的任一属性值，因为 s 为 r 的唯一个分解，故必有 $k_{ij} \subseteq v_{ij}$，因而 $\alpha_i \subseteq t_i$，

$\alpha_i \subseteq \underline{R}\, t_i$，另外对于 r 来说，仅有 $\alpha_i \subseteq \underline{R}\, t_i$，因此 $\underline{R}\, t_i = \alpha_i$ 成立；同理，对于任意的 $j \neq i$，有 $\alpha_j \subseteq \underline{R}\, t_j$，$\underline{R}\, t_j = \alpha_j$，按照粗糙集原理，有 $\underline{R}\, r = \{\underline{R}\, t_1, \underline{R}\, t_2 \cdots \underline{R}\, t_n\} = \{\alpha_1, \alpha_2, \cdots, \alpha_n\} = s$；对于任意的 t_1，有 $t_i \cap t_i = t_i \neq \phi$，因此 $t_i \subseteq \overline{R}\, t_i$，且仅有 $\overline{R}\, t_i = t_i$，依次类推，$\overline{R}_r = \{\overline{R}_{t_1}, \overline{R}_{t_2} \cdots \overline{R}_{t_n}\} = \{t_1, t_2, \cdots, t_n\} = r$；该定理得证。

定理 2.2　设 s 为粗糙关系 r 的任一分解，则必有 $\overline{R}\, s = r$。

推论 2.1　给定 s 为粗糙关系 r 的任一分解，s 由元组集合（α_1，α_2，…，α_n）构成，r 由元组集合（t_1，t_2，…，t_n）组成，k_{ij} 为 s 的任一元组 $\alpha_i (1 \leqslant i \leqslant n)$ 的任一属性值，v_{ij} 为 r 的任一元组 t_i 的任一属性值，$\forall\, \alpha_i$，若 $\exists\, k_{ij} \subset v_{ij}$，则 $\underline{R}\, s = \phi$。

证明：设 s 为粗糙关系 r 的任一分解，它由元组集合（α_1，α_2，…，α_n）构成，k_{ij} 为 s 的任一元组 $\alpha_i (1 \leqslant i \leqslant n)$ 的任一属性值，v_{ij} 为 r 的任一元组 t_i 的任一属性值，$\forall\, \alpha_i$，若 $\exists\, k_{ij} \subset v_{ij}$，则按照上、下近似算子的定义，$t_i \in s$ 不成立，即没有一个元组是 s 的元素，故 $\underline{R}\, s = \phi$。

推论 2.2　粗糙关系 s 为粗糙关系 r 的分解的充要条件是 $k_{ij} \subseteq v_{ij}$（其中 k_{ij} 为 s 的任一元组 $\alpha_i (1 \leqslant i \leqslant n)$ 的任一属性值，v_{ij} 为 r 的任一元组 t_i 的任一属性值）。

例如，表 2-2 为表 2-1 的一个多值分解。

表 2-1　子域

ID	COUNTRY	FEATURE
U123	US	{MARSH，LAKE}
U124	US	MARSH
U125	USA	{MARSH，PASTURE，RIVER}
U126	US	{FOREST，RIVER}
U147	US	{SAND，ROAD，URBAN}
U157	{US，MEXICO}	{SAND，ROAD}
M007	MEXICO	{SAND，ROAD}

续表

M008	MEXICO	BEACH
M009	MEXICO	SAND
CO39	BELIZE	JUNGLE
CO40	{BELIZE，INT}	{JUNGLE，COAST，SEA}

表 2-2　分解模式

ID	COUNTRY	FEATURE
U123	US	{MARSH }
U124	US	MARSH
U125	USA	{MARSH，PASTURE}
U126	US	{FOREST}
U147	US	{SAND}
U157	{US}	{SAND，ROAD}
M007	MEXICO	{SAND，ROAD}
M008	MEXICO	BEACH
M009	MEXICO	SAND
CO39	BELIZE	JUNGLE
CO40	{BELIZE，INT}	{JUNGLE}

2.4　RRDB与FRDB关系的系统研究

本节以粗糙集理论和关系数据库理论为基础，从函数依赖、范式理论和 Armstrong 公理等方面系统地研究了粗糙关系数据库（Rough Relational Database，RRDB）与模糊关系数据库（Fuzzy Relational Database，FRDB）之间的关系。结果表明，模糊函数依赖与粗糙函数依赖均为经典函数依赖的泛化，模糊范式理论为经典范式的扩充，而粗糙范式理论自成体系，从推理规则上看，

它们都不同程度地符合 Armstrong 公理[120]。

2.4.1　引言

关系数据模型是由 E.F.Codd 于 1970 年开始发表的系列论文提出的，目前，关系数据库理论的研究逐渐趋于成熟。关系数据模型处理的主要是具有良好定义的、清晰明确的数据，而现实世界中的数据通常是含糊的、不完全的和不确定的，因此，人们又提出了各种处理含糊数据的扩充的数据库模型，其中，粗糙关系数据库与模糊关系数据库是处理含糊和不确定数据的两种重要的数据库。

Zadeh 提出的模糊集理论是精确集合论的泛化，而模糊关系则是关系概念的泛化，由 Buckle 和 Petry 提出的模糊关系数据库是建立在相似关系的基础上的，是对经典关系数据库的泛化。而粗糙关系数据库是由 Theresa Beaubouef 与 Petry 于 1993 年提出的，该数据库模型把粗糙集的本质特征结合到基本的关系数据库中，极大地改善了关系数据库的查询性能，它同样可以处理不确定的知识。那么它与模糊关系数据库有什么联系呢，本节将从其基本概念、函数依赖、数据库范式和 Armstrong 公理的符合情况几方面对它们的关系进行系统的研究。

2.4.2　FRDB与RRDB的概念分析

Buckle 和 Petry 提出的模糊关系数据库是把模糊信息结合到关系数据库结构中形成的，像经典的关系数据库理论一样，FRDB 定义为关系的集合，而其每个关系为元组的集合。设用 t_i 表示第 i 个元组，那么 $t_i=(d_{i1}, d_{i2}, \cdots, d_{im})$，在经典的关系数据库中，$d_{ij}\in D_j$，其中 D_j 为值域，FRDB 与关系模型最大的不同是 $d_{ij}\subseteq D_j$，即其属性值可以为一子集[127]。

这里定义的 FRDB 是基于相似关系的，下面给出 FRDB 的基本概念。

定义 2.26[127] 模糊数据库 R 定义为集合叉积 $2^{D1}\times 2^{D2}\times\cdots\times 2^{Dm}$ 的一个子集，这里 $2^{Dj}=2^{Dj}-\phi$。设 $R\subseteq 2^{D1}\times 2^{D2}\times\cdots\times 2^{Dm}$，模糊元组 t 为 R 的一个元素。

上述的 FRDB 的定义是依托于相似关系的（Similarity Relation），下面给出其定义。

定义 2.27[127] 相似关系定义为映射 $S_j:D_i\times D_j\rightarrow[0，1]$，对于 x，y，$z\in D$ 有：

$S_j(x，x)=1$ (自反性)

$S_j(x，y)=S_j(y，x)$ (对称性)

$S_j(x，z)\geqslant \max\limits_{y\in D_j}\{\min[S_j(x，y)，S_j(y，z)]\}$（最小—最大传递性）

Theresa Beaubouef 在研究粗糙集的基础上把粗糙集与数据库结合起来提出了粗糙关系数据库模型（简称 RRDM），并定义了各种粗糙关系操作算子。RRDM 将粗糙集的重要性质引入到基本关系模型中，从而使之具有更好的检索能力和适应性。

把 RRDM 支持的数据库称为粗糙关系数据库（Rough Relational Database，RRDB）。从 FRDB 与 RRDB 的定义可以得出：

（1）二者均为关系数据库的泛化。FRDB 通过模糊关系完成对普通关系的泛化，而 RRDB 则通过粗糙关系泛化普通关系。从本质上讲，它们均扩充了普通关系，是可以属于非 1NF（第一范式）的数据库，这点从 $d_{ij}\subseteq D_j$ 可以看出（FRDB 可以满足 1NF，分为 1 型，2 型）。

（2）FRDB 建立在相似关系基础上，而 RRDB 建立在等价关系基础上。它们均具有自反性和对称性，但前者只具有最小—最大传递性，而后者则具有经典的传递性。

（3）二者均可对关系进行划分，但其分类思想不同。相似关系泛化了等价关系，每个相似关系均可形成一个划分，其关联着模糊隶属值，因此 FRDB 从某种程度上保留了关系数据库的一些重要特征；而 RRDB 的划分则是建立在等价关系上的。换句话

说，FRDB 对关系的分类建立在元素对集合的隶属度上，RRDB 对于关系的划分则建立在对象之间是否等价上。

2.4.3　模糊函数依赖（FFD）、粗糙函数依赖（RFD）与Armstrong公理

在文献[127]中，Shenoi 给出了模糊函数依赖的定义，其定义是以相似关系为基础的。

定义 2.28　设 D 为一个确定论域，R 为定义在 $D\times D$ 上的相似关系，称 x、$y\in D$ 为 α 级相似，当且仅当 $R(x, y)\geqslant\alpha$。也把它称为 α 级冗余，记为 $x\sim\alpha y$。

定义 2.29　对于模糊关系数据库 $R=\{A_1, A_2, \cdots, A_n\}$，设其任意的两个元组是 α 级冗余的（记为 $\sim\alpha$），t_1，t_2 为 R 的任意两个元组，X、$Y\subseteq R$ 为 R 的属性子集，X，Y 之间的模糊函数依赖 $X\xrightarrow[\alpha X,\alpha Y]{}Y$ 成立，当且仅当 $t_1(X)\sim\alpha X\, t_2(X)$ 时，有 $t_1(Y)\sim\alpha Y\, t_2(Y)$，其中 $\sim\alpha X$、$\sim\alpha Y$ 均为 α 级冗余。

设 $t_1=(d_{x1}, d_{x2}, \cdots, d_{xm})$，$t_2=(d_{y1}, d_{y2}, \cdots, d_{ym})$ 为 RRDB 的任意两个子元组（子元组可以看作是元组本身或元组的一部分），对任意的 j=1，2，…，m，当 $[d_{xj}]=[d_{yj}]$ 时，t_1，t_2 是元组冗余（Redundant）的，其中 $[d]$ 表示包含 d 的等价类。

定义 2.30[42]　（粗糙冗余）设 $t_1=(d_{x1}, d_{x2}, \cdots, d_{xm})$，$t_2=(d_{y1}, d_{y2}, \cdots, d_{ym})$ 为 RRDB 的任意两个子元组，对任意的 j=1，2，…，m，当存在 p、q 使得 $[p]\subset[d_{xj}]$，$[q]\subset[d_{yj}]$ 时，有 $[p]=[q]$，则称 t_1，t_2 是粗糙冗余（Rough Redundant）的。

定义 2.31[42]　设 X，Y 为粗糙关系模式 R 的属性子集，粗糙函数依赖（RFD）$X\rightarrow Y$ 对于一个粗糙关系模式 R 的所有实例 T 都成立，当满足：

（1）对任意两个元组 t，$t'\in\underline{R}\,T$，Redundant($t(X)$，$t'(X)$)→

Redundant($t(Y)$，$t'(Y)$)且

（2）对任意两个元组 s，$s' \in \overline{R}\,T$，Rough-redundant($s(X)$，$s'(X)$)→Rough-redundant ($s(Y)$，$s'(Y)$)。

比较 RRDB 包含的概念，如粗糙关系、RRDM、粗糙冗余、粗糙元组与 FRDB 的模糊关系、FRDM、模糊冗余和模糊元组等，可以看出，它们的定义极为相似，可以认为，RRDB 的各种概念的定义是由 FRDB 的相关概念引申而来的，RRDB 所处理的问题实际上是 FRDB 所处理问题的补充与完善。因此可以得出结论：

（1）首先，RFD 与 FFD 的应用背景不同。根据定义，RFD 是用于支持 RRDB 的，而 FFD 则既可用于关系数据库，亦可用于 FRDB。

（2）RFD 与 FFD 均是普通函数依赖的泛化。FFD 利用相似关系减弱等价性来泛化函数依赖，RFD 利用不可分辨关系与相似关系来泛化函数依赖，函数依赖是 RFD 和 FFD 的特例，在特殊的情形下，RFD 和 FFD 均可转化为普通的函数依赖（RRDB 变为 RDB 时，则 RFD 变为普通函数依赖；FFD 中相似值均取 1 则可转化为普通的函数依赖）。

（3）RFD 与 FFD 同普通的函数依赖一样均满足 Armstrong 公理及其附加的推理规则（FFD 及普通函数依赖满足 Armstrong 公理及其推理规则已被证明，后面还要分析）。

（4）RFD 与 FFD 泛化函数依赖的形式不同：RFD 利用粗糙集理论的上、下近似与谓词描述来定义，FFD 则利用相似关系等来扩充函数依赖，从对等价性的要求看，RFD 比 FFD 要更宽松一些，FFD 要求具有函数依赖的属性对应的属性值具有同等的相似度，而 RFD 则利用等价类与 Rough 冗余来定义，从本质上讲，RFD 定义中的（1）实际上是等同于经典的函数依赖的，而其定义中的（2）则要求具有函数依赖的属性对应的属性值包含相同的冗余度[112]。

对于 FRDB 其函数依赖满足下列 Armstrong 公理推理规则[127]：

模糊包含规则：如果 $X \xrightarrow{\theta_1}{}_{\mathrm{F}} Y$ 成立，且$\theta_1 \geqslant \theta_2$，那么 $X \xrightarrow{\theta_2}{}_{\mathrm{F}} Y$ 成立；

模糊自反规则：如果 $X \supseteq Y$，那么 $X \rightarrow_{\mathrm{F}} Y$ 成立；

模糊增广规则：$\{X \xrightarrow{\theta}{}_{\mathrm{F}} Y\}|=\mathrm{XZ} \xrightarrow{\theta}{}_{\mathrm{F}} \mathrm{YZ}$；

模糊传递规则：$\{X \xrightarrow{\theta_1}{}_{\mathrm{F}} Y,\ Y \xrightarrow{\theta_2}{}_{\mathrm{F}} Z\}|=X \xrightarrow{\min(\theta_1,\theta_2)}{}_{\mathrm{F}} Z$。

并且，模糊函数依赖满足下列 Armstrong 公理附加推理规则：

模糊合并规则：$\{X \xrightarrow{\theta_1}{}_{\mathrm{F}} Y, X \xrightarrow{\theta_2}{}_{\mathrm{F}} Z\}|=X \xrightarrow{\min(\theta_1,\theta_2)}{}_{\mathrm{F}} \mathrm{YZ}$；

模糊伪传递规则：$\{X \xrightarrow{\theta_1}{}_{\mathrm{F}} Y,\ \mathrm{WY} \xrightarrow{\theta_2}{}_{\mathrm{F}} Z\}|=\mathrm{WX} \xrightarrow{\min(\theta_1,\theta_2)}{}_{\mathrm{F}} Z$；

模糊分解规则：如果 $X \xrightarrow{\theta}{}_{\mathrm{F}} Y$ 成立，且 $Z \supseteq Y$，那么 $Y \xrightarrow{\theta}{}_{\mathrm{F}} Z$ 成立。

对于 RRDB，在文献[112]中已经证明了其同样满足 Armstrong 公理推理规则：

设 U 为属性集总体，F 是 U 上的一组粗糙函数依赖，于是有粗糙关系模式 $R<U, F>$，对于 $R<U, F>$来说有以下粗糙函数依赖推理规则（Armstrong 公理）：

RFD1：自反律若 $Y \subseteq X \subseteq U$，则 $X \xrightarrow{R} Y$ 为 F 所蕴涵。

RFD2：传递律若 $X \xrightarrow{R} Y$，$Y \xrightarrow{R} Z$ 为 F 所蕴涵，则 $X \xrightarrow{R} Z$ 为 F 所蕴涵。

RFD3：增广律若 $X \xrightarrow{R} Y$ 为 F 所蕴涵，且 $Z \subseteq U$，则 $XZ \xrightarrow{R} YZ$ 为 F 所蕴涵。

对于 Armstrong 公理附加推理规则，RRDB 同样满足：

RFD4：合并规则由 $X \xrightarrow{R} Y$，$X \xrightarrow{R} Z$，有 $X \xrightarrow{R} YZ$；

RFD5：分解规则由 $X \xrightarrow{R} Y$ 及 $Z \subseteq Y$，有 $X \xrightarrow{R} Z$；

RFD6：伪传递规则由 $X \xrightarrow{R} Y$，$\mathrm{WY} \xrightarrow{R} Z$，有

$XW \xrightarrow{R} Z$。

比较模糊函数依赖与粗糙函数依赖对 Armstrong 公理及附加的推理规则的适合情况，可以看出，虽然它们均满足 Armstrong 公理及附加推理规则，但是它们所符合的条件和原理是不同的。FFD 对于 Armstrong 公理及附加推理规则的满足是以相似度为基础的，且模糊传递规则只是满足最小传递，并非经典关系数据库意义下传递。模糊合并规则、模糊伪传递规则也同样如此，从一定意义上讲，这是一种弱传递。而粗糙函数依赖则不同，从其定义可知，RFD 是定义于粗糙集原理的上近似、下近似上的，而 $\underline{RT}$ 、$\overline{RT}$ 显然是以经典的等价关系为基础的，虽然 RRDB 是非 1NF 的，因而可以得出，粗糙函数依赖对于 Armstrong 公理及附加推理规则的满足是以粗糙集为基础的，其满足 Armstrong 公理及附加推理规则的程度要更大一些。

另外，FFD 对 Armstrong 公理经典三规则的符合上有一点值得强调，就是增加了模糊包含规则：如果 $X \xrightarrow{\theta_1}_{\mathrm{F}} Y$ 成立，且 $\theta_1 \geqslant \theta_2$，那么 $X \xrightarrow{\theta_2}_{\mathrm{F}} Y$ 成立，这是 FFD 所特有的。

2.4.4 FRDB与RRDB的范式

在经典的关系数据库中定义范式是为了指导设计关系数据库，避免出现不希望出现的情形，如修改异常和数据不一致等，而类似的问题在模糊关系数据库中同样存在。因此，基于 FFD，人们同样研究 FRDB 对于范式的满足情况（定义 2.32～2.35 参考文献[127]）。

定义 2.32 模糊关系模式 R 具有属性集 U 和划分级 α_{U} 满足第一模糊范式。

经典关系理论的第二、第三范式是用于处理数据库关系的主码与非码属性的关系的。与关系数据库相似，FRDB 的第二、第三范式可以利用 FFD 来定义。

定义 2.33　对于具有属性集 U 和划分级 α_U 的模糊关系模式 R，当 R 满足第一模糊范式且其每个非码属性完全地依赖于 R 的每个键（对应于划分级 α_U），其满足第二模糊范式。

关系数据库模型中第三范式是用于规范主码与非码属性间的传递依赖的。

定义 2.34　对于具有属性集 U 和划分级 α_U 的模糊关系模式 R，当 R 满足第一模糊范式且没有非码属性传递依赖于 R 的主码（对应于划分级 α_U），则 R 满足模糊第三范式。

关系模型中的 BCNF 范式是规范化程度最高的范式（单值环境下），对于 FRDB，同样可以定义模糊 BCNF。

定义 2.35　对于具有属性集 U 和划分级 α_U 的模糊关系模式 R，当 R 满足第一模糊范式且没有属性传递依赖于 R 的主码（对应于划分级 α_U），则 R 满足模糊 BCNF。

可以看到，上述 FRDB 范式的定义是以划分级 α_U 为基础的，模糊范式的定义通过泛化冗余的概念扩展经典的关系范式，把冗余的模糊概念表示为相同等价类非空子集的等价，替代经典的元素等价，从这点可知，FRDB 的范式为关系模型范式的泛化。

再来分析 RRDB 的范式情况。Theresa Beaubouef 等人在文献[97]中提出了 RRDB 的粗糙第二范式、粗糙第三范式及粗糙 BCNF，值得一提的是 RRDB 不存在 1NF，因为它本身是非第一范式的（定义 2.36～2.38 参考文献[128]）。

定义 2.36　设 F 为模式 R 的粗糙函数依赖集，且 K 为 R 的主码，那么 R 满足粗糙 2NF，当且仅当没有非主属性部分粗糙依赖于 K。

定义 2.37　设 F 为模式 R 的粗糙函数依赖集，且 K 为 R 的主码，那么 R 满足粗糙 3NF 当 R 没有传递依赖 $G \rightarrow H$ 存在，这里或者 G 为超码，或者 H 为主属性。

定义 2.38　设 F 为模式 R 的粗糙函数依赖集，且 K 为 R 的

主码，那么 R 满足粗糙 BCNF 当 R 满足粗糙 3NF 且 F 中没有传递依赖 $G \rightarrow H$ 存在，这里 G 为超码。

从 FRDB 与 RRDB 的范式定义分析可知，FRDB 的 2NF，3NF 和 BCNF 的定义是以 1NF 为基础的，并且依赖于隶属度为基础的划分级α_U，而 RRDB 由于其本身为非 1NF 的，因此无法按照传统的 1NF 来定义，因此 RRDB 的 2NF，3NF 和 BCNF 的定义无法以 1NF 为基础，只能部分按照经典范式来定义。但总体来说，FRDB 与 RRDB 的定义均可以看作经典范式的泛化。另一方面，由于 RRDB 的范式理论是 Theresa Beaubouef 等于 2005 年提出的，理论本身并不成熟，且由于 RRDB 本身是非 1NF 的，因此它的许多概念包括范式并不能简单地由关系模型照搬，因此 RRDB 的范式理论应该说还很不成熟甚至有不妥当的地方，这方面还有待深入研究。

第 3 章　粗糙数据查询

吾生也有涯，而知也无涯

——《庄子·养生主》

3.1　引言

数据查询是数据库管理系统的基本功能之一，数据库的查询就是从数据库中检索出满足查询条件的元组。现有的许多关系数据库的查询语言（包括 SQL 语言）都基于下列方式之一：关系代数（Relational Algebra）和关系演算（Relational Calculus）。关系代数为关系定义了一系列算子，查询表达为一个算子序列（复合）；关系演算是基于一阶谓词演算，查询表达为关系元组必须满足的谓词。尽管作为查询语言的这两种基础形式不同，但它们一般是等价的（在安全限制的条件下）。

一般地，可以把数据查询分为：普通数据查询、模糊数据查询（包括概率模糊查询）和粗糙数据查询。

Beaubouef Theresa Ann 等人在文献[42]中提出了粗糙关系数据库模型（RRDM），并对它进行了定义，对粗糙关系操作包括并、差、选择、投影、连接及相关性质等进行了研究。胡可云和眭跃飞等在文献[96]中对粗糙关系数据库（RRDB）的粗糙关系的上、下近似给出了几种定义，并对熵规则进行了分析。从目前 RRDB 的研究情况来看，大多还停留在理论研究阶段，而 RRDB 的应用研究则比较少。Beaubouef Theresa Ann 研究了精确数据的粗糙查询，但是并没有系统地研究关于 RRDB 的查询方法。

本章以粗糙关系数据库模型为背景，从分解原理、投影原理和粗糙关系数据库（Rough Relation Database，RRDB）的可定义性几方面讨论了 RRDB 的查询机理，并以此为基础研究了基于粗糙集理论的 RRDB 的查询方法。把粗糙数据查询分为精确查询、粗糙完全查询和粗糙组合查询三类，并从这三方面对粗糙数据查询进行了讨论与仿真实验，仿真结果验证了本文查询思想的可行性和正确性。

定义 3.1[129] 确定性信息系统（Deterministic Information Systems，DIS）DIS =（OB，AT，{$\mathrm{VAL}_a|a\in AT$，f}），其中 OB 是对象组成的有限集合，AT 是属性的有限集合，VAL_A 是属性值组成的有限集合，f 为映射 f：OB×AT→∪VAL 称为分类函数。

定义 3.2[129] 一个不确定性信息系统（Non-deterministic Information Systems，NIS）是一个四元组：$(OB,\mathrm{AT},\{\mathrm{VAL}_A\,|\,A\in \mathrm{AT}\},g)$，其中 OB 是对象组成的有限集合，AT 是属性的有限集合，VAL_A 是属性值组成的有限集合，$g:OB\times AT\to P(\bigcup_{A\in AT}\mathrm{VAL}_A)$ 的映射，$P(\bigcup_{A\in AT}\mathrm{VAL}_A)$ 是指 VAL_A 上的幂集。如果不知道确切的属性值，则 $g(x,A)=\mathrm{VAL}_A$。

粗糙集处理的表是确定性信息，也称为确定性信息系统。不确定性信息系统与不完备信息系统（Incomplete Information Systems）已经被提出用于研究 DIS 的不完备性，例如空值、未知值和误值等[130]。

日本学者 Hiroshi Sakai 与 Akimichi Okuma 对不确定性信息系统进行了多年的研究，分别就 NIS 的以下问题进行了研究：

（1）NIS 中集合的可定义性及其在计算机中的实现算法；

（2）NIS 中对象的一致性及其在计算机中的相关实现算法；

（3）NIS 中数据依赖性及其在计算机中的相关实现算法；

（4）NIS 中规则提取及其在计算机中的相关实现算法；

（5）NIS 中属性约简及其在计算机中的相关实现算法。

虽然 RRDB 与 NIS 在形式上是相同的，但是他们研究的问题却不一样，本章对 RRDB 与 NIS 的关系的关系问题进行了简单探讨。

3.2　数据库查询思想

3.2.1　查询与模糊查询

数据查询是数据库的最基本、最常用和最复杂的操作之一，可以说数据库的查询是数据库的核心操作。数据库的查询一般都以查询语言表示，如用 SQL 表示。从查询语句出发，直到获得查询结果，需要一个处理过程，这个过程叫做查询处理。关系数据库的查询语言一般是非过程性语言，仅表达查询的要求，而不说明查询的过程，也就是用户不必关心查询语言的具体执行过程，而由 DBMS 确定合理的、有效的执行策略，DBMS 在这方面的作用称为查询优化，对于配有非过程查询语言的 DBMS，查询优化是查询处理过程中的重要和必要的一环，对系统的性能影响很大[131]。

数据库查询语言的处理过程与一般高级语言相似，先做词法分析和语法分析，把查询语句变为语法树或语法图，然后进行查询优化，形成执行计划，生成可执行代码，交系统执行。

除普通的关系数据库的数据查询外，常用的查询思想还有模糊数据查询和粗糙数据查询。

对于数据库的检索，不仅有精确匹配的要求，还要求对相似概念和相近概念的检索，这就需要建立模糊数据库，以便进行含糊数据的检索。有多种不同的含糊性或不确定性，如边界不确定的含糊性、不完备信息的含糊性、数据不一致的含糊性等，模糊数据库的最重要任务之一就是把咨询和存储的数据之间的匹配由{0，1}扩充到闭区间[0，1]上，即在匹配中除了相等和不相等外，还引入了相似的概念，其特征函数可以取 0 与 1 之间的任意

值，从而得到了描述模糊集合的特征函数——模糊隶属函数。为了使数据库能够处理现实世界中广泛存在的不精确或模糊信息，Buckles 等人于 1977 年建立了模糊关系数据库理论关系模糊，数据库可以把含糊性引入到存储数据、管理系统、咨询术语和关系模型等方面，这一理论近几十年来得到了迅速地发展[132~133]。

所谓模糊查询指只要输入指定信息的任意一个含糊值，就可将所需信息查找出来。依据查询对象的不同，数据库的模糊查询分为传统关系数据库的模糊查询与模糊关系数据库的模糊查询。

对传统的关系数据库进行模糊查询时，只涉及查询条件中存在模糊量和模糊运算符，而查询的对象是精确值集合。对于任意的模糊关系数据库，都可利用将模糊量转变成精确值的方法，将其转变成精确的数据库，这个精确数据库是原模糊关系数据库的像，一般可以利用映射函数将对模糊关系数据库的查询转变为对模糊关系数据库的像的查询。

3.2.2 粗糙数据查询

粗糙数据查询也称粗糙集查询（Rough Set Querying）。可以把粗糙集理论的等价类和粗糙谓词（Rough Predicates）与查询理论相结合以形成粗糙查询。粗糙数据查询可以分为关系数据库的粗糙查询和粗糙关系数据库的粗糙查询，Beaubouef Theresa Ann 主要研究了关系数据库的粗糙查询（采用谓词来研究），而对粗糙关系数据库的粗糙查询并没有做深入的研究。

Beaubouef Theresa Ann 认为对于关系数据库进行粗糙数据查询需要做一些必要的数据规范，但是不改变数据库中实际的应用数据[42]。

首先需要指定的规范是不可分辨关系。对于一个不可分辨关系来说，它能够将指定的域进行等价类划分。对于精确数据（关系数据库）来说，它们只属于包含它自身的等价类，如果在指定的论域上没有进行不可分辨关系（等价关系）定义，那么该论域

被认为是精确的且自动划分为只包含数据本身的若干个等价类。

在用户定义好不可分辨关系后，粗糙谓词作为未来查询的一部分被用户定义。这些谓词与所有的查询一起使用预先定义的不可分辨关系，谓词与用户定义的属性值的上、下近似或属性值的逻辑组合相关联，用以定义一个作为结果元组的粗糙集。谓词的下近似表示用户属性值或元组可以确定地满足相应的概念的条件，上近似表示用户属性值或元组可能满足相应的概念的条件，用户必须说明下近似及边界域需求，这样谓词作为数据库语义的一部分直接在查询中使用，一般谓词具有下面的形式：

Property(Relation Name)={t|满足下近似的条件}

{t|满足边界域的条件}。

从以上的描述可以看出，所谓粗糙查询是利用不可分辨关系进行数据库检索的操作。实际上，粗糙集理论中的下近似定义给出了被查询文档或相关查询的可定义的特性，而上近似则给出了相关文档或查询的可能被描述的特性，这一特点为粗糙集可以应用于数据查询奠定了理论基础。

从查询语言看，关系数据库中普通查询使用的查询语言为 SQL，而模糊查询则使用用户扩充的 FSQL，文献[42]中对于精确数据的粗糙查询使用了作者定义的 RSQL 语言。

除上述的基本查询思想外，另外还有一些查询方法是把相关学科与它们相互组合形成的一些思想，如概率模糊查询和模糊粗糙查询等。

下面将利用粗糙集来分析粗糙关系数据库的查询原理与操作。

3.3　RRDB的分解与投影原理

3.3.1　RRDB的分解原理

要对粗糙关系数据库进行数据查询，必然涉及到的粗糙关系

数据库某个关系的部分数据，具体地说是，就是要涉及到某个元组或其中的一部分，因此在 Beaubouef Theresa Ann 定义的“解释”概念的基础上定义“子解释”，从对元组的分解进而引申到对粗糙关系数据库关系（由许多粗糙关系构成）的分解。

定义 3.3[110] 把 RRDB 定义为三元组：$S=(U, A, D_i)$，从数据库的角度看，U 是所有元组的集合，A 为数据库的属性集，D_i 为某一属性值域，对于每个 $r\in U$，$a\in A$，有 $r(a)\subseteq D_a$，$r(a)$为 r 在属性 a 上的取值。按照信息系统的定义，RRDB 实际上是特殊的多值信息系统，其中：$U=\{u_1, u_2, \cdots, u_{|U|}\}$是对象的全体，它是非空有限集，即论域；$A=\{a_1, a_2, \cdots, a_{|A|}\}$是属性的全体，它是非空有限集。

定义 3.4[110] （子解释）粗糙元组 $t_i=(d_{i1}, d_{i2}, \cdots, d_{im})$的解释 $\alpha=(a_1, a_2, \cdots, a_m)$：对 $\forall j(1\leqslant j\leqslant m)$来说，是进行值的分配，即使 $a_j\in d_{ij}$，称 α 为 t_i 的解释，称 a_j 为 d_{ij} 的子解释。

在 RRDB 中数据查询中，用Γ表示分解算子。

3.3.2 RRDB的投影原理

粗糙关系数据库的投影原理与关系数据库的投影原理是类似定义的。

投影原理[110]：其基本思想是从一个关系中选择所需要的属性并按要求排列组成一个新的关系，新关系的各个属性值来自原关系中相应的属性值，并去掉重复元组。对于一个粗糙关系 X，它在属性 B 上的投影 $\prod_B(X)=\{t(B)|t\in X\}$。

粗糙关系数据库的投影原理也可以利用马垣老师给出的定义来说明[105]。

在 RRDB 中，把投影和分解算子结合起来，可以给出解释和子解释新的形式化描述：

粗糙元组 $t_i=(d_{i1}, d_{i2}, \cdots, d_{im})$的一个解释$\alpha=(a_1, a_2, \cdots, a_m)$可以表示为：

$\alpha=(\Gamma_{k1}(r_i(\prod(at_1))),\ \Gamma_{k2}(r_i(\prod(at_2))),\ \cdots,\ \Gamma_{km}(r_i(\prod(at_m))))$

其中，$r_i(\prod(at_1))\in d_{i1}$；$r_i(\prod(at_2))\in d_{i2}$；$r_i(\prod(at_m))\in d_{im}$；$at_i$ 为某一属性。单值分解的元组便可以表示为上述的 α 形式。

3.3.3　RRDB的可定义性

所谓 RRDB 的可定义性，指的是如何用粗糙集的上、下近似算子和粗糙关系数据库的元组表示相应的查询结果。

按照 Hiroshi Sakai 的观点[129]，多值信息系统实际上是一种不确定信息系统，因而它应该符合 M.Kryszkiewicz 的关于不完备信息系统的上、下近似的定义：

$$\underline{A}\,\mathrm{SIM}(X)=\{x\in\theta\,|\mathrm{SA}(x)\subseteq X\},\quad \overline{A}\,\mathrm{SIM}(X)=\{x\in\theta\,|\mathrm{SA}(x)\cap X\neq\phi\}$$

其中，SIM（A）为相似关系，$\mathrm{SIM}(A)=\{(x,\ y)\in U\times U|\forall\ a\in A,\ a(x)=a(y)$ or $a(x)=*$ or $a(y)=*\}$，$A\subseteq\mathrm{AT}$；$\mathrm{SA}(x)=\{y\in U|(x,\ y)\in\mathrm{SIM}(A)\}$。对于 A 而言，$SA(x)$是与 x 可能不可区分的对象的最大集合[23]。

粗糙关系数据库虽然是多值信息系统的一种，但是它同时又具有数据库的某些特性，因而上述定义对于进行 RRDB 的数据查询并不完全适合，因此需要研究 RRDB 中被查询集合的粗集表示。

定义 3.5[96]　设 $S=(U,\ A,\ D_i)$为 RRDB，对于属性 $a\in A$，定义 U 上的共同关系 θ_a，$r\,\theta_a\,s\Leftrightarrow r(a)\cap s(a)\neq\phi$，给定属性集 $B\subseteq A$，定义 B 上的共同关系 θ_B 为 $\bigcap_{a\in B}\theta_a$。

定义 3.6　设 θ_a 为 RRDB 上的共同关系，则对于 $u\in U$，其共同类 $[u]_{\theta_a}=\{v\in U|v(a)\ \cap u(a)\neq\phi\}$，对于 v_1，$v_2\in U$，v_1，v_2 在同一个类 $[u]_{\theta_a}$ 中，当且仅当 $v_1(a)\cap v_2(a)\ \cap u(a)\neq\phi$，它满足：①若 $[u]_{\theta_a}\cap[v]_{\theta_a}=\phi$，那么 $u\,\theta_a\,v$ 不成立；②$\bigcup_{u\in U}[u]_{\theta_a}=U$。

定义 3.7[110]　对于一个具有模式 R 的 RRDB 来说，设 $X\in U$ 为对于 RRDB 进行粗糙查询操作的集合，当 X 不能用 R 中的元

组表示时，称 X 在 R 中是不可定义的，这时 $\overline{R}X=\underline{R}X=\phi$，查询结果为空；当 X 可以用 R 中的元组精确表示时，称 X 在 R 中是精确可定义的，这时 $\overline{R}X=\underline{R}X=\{r_i|r_i\in R\wedge r_i\in X，1\leqslant i\leqslant|U|\}$；当 X 可以用 R 中的元组表示但不能精确表示时，称 X 在 R 中是粗糙可定义的，这时有：

$$\overline{R}X=\{r_i|\exists i(r_i\in R)\wedge|r_i（a）|\geqslant 1\wedge r_i\cap X\neq\phi，1\leqslant i\leqslant|U|\}$$

$$\underline{R}X=\{r_i|r_i\in R\wedge|r_i（a_j）|=1\wedge r_i\in X，1\leqslant i\leqslant|U|，1\leqslant j\leqslant|A|\}$$

其中，r_i 表示 R 的某个元组；$|r_i（a_j）|$表示某个属性值 $r_i（a_j）$的子解释数；a_j 为某个属性。设 RRDB 的任一个属性值为（$r_i（a_{j1}）$，$r_i（a_{j2}）$，…，$r_i（a_{jk}）$），其中，$r_i（a_{j1}）$，$r_i（a_{j2}）$，…，$r_i（a_{jk}）$是 $r_i（a_j）$的 k 个子解释。本文主要研究属性值的子解释间的关系为“或”的情况，即其 k 个子解释的关系为“$\vee$”的情况，若其子解释间的关系为“与”，即其子解释间的关系为“$\wedge$”，这时可以认为$|r_i（a_j）|=|r_i（a_{j1}）\wedge r_i（a_{j2}）\wedge\cdots\wedge r_i（a_{jk}）|=1$。

3.4 RRDB的粗糙数据查询

对于粗糙关系数据库来说，由于它的属性值具有多值，因此对它查询时需要进行相应变换，本书的处理方法是根据共同关系或等价关系将粗糙关系数据库分解，然后再用 SQL 语言和粗糙关系操作对它进行处理，这就是本文提出的针对粗糙关系数据库的粗糙数据查询。

3.4.1 精确查询

所谓对粗糙关系数据库的精确查询，是一种与检索关键字完全匹配的数据查询。

对粗糙关系数据库进行精确查询，采用粗糙集的等价关系原理，其方法如下。

（1）对于要查询的数据，分析其等价类（含语义相同的属性值），然后根据所得到的等价类及要查询的字段值将 RRDB 分解为 RRDB 的一个子表，该子表中要查询的字段值及检索字段全部保留。

（2）分析得到的子表，若其为单值分解，则利用 SQL 实现数据查询，若其为多值分解，则利用粗糙关系操作（如投影）完成数据查询。

采用上述方法把对粗糙关系数据库的查询转化为对它的分解的查询，之所以要形成该子表，是为了保留中间结果以辅助相关操作。

以表 3-1[42]为例，查询所有国家为 US 的所有 ID。

根据，“US”与“USA”属于同一个等价类，因此上述要求可以表示为：

$\Pi_{1,2}$（$\sigma_{\text{COUNTRY=“US” or COUNTRY=“USA”}}$（SUBREGIONS））

从表 3-1 可以抽取 1, 2 列形成表 3-2（子表 1），其记录满足条件：r_i(COUNTRY)=“US”或 r_i(COUNTRY)=“USA”。上面得到的子表为一单值分解，因此利用 SQL 实现数据查询

```
SELECT   ID
FROM   SUBTABLE1
WHERE   COUNTRY=“US”  OR COUNTRY=“USA”;
```

查询结果为{U123，U124，U125，U126，U147}，在 SELECT 语句中，实际上是选择 ID 这样一个等价类，而 WHERE 语句中，实际上是以 US 的等价类[US]为选择条件的，这样，便实现了 RRDB 的基于等价关系的精确查询。

定理 3.1　精确查询的结果为满足查询条件的最小集合，它的结果 X 可以表示为 $X=\overline{R}\,X=\underline{R}\,X=\{r_i|r_i\in R\wedge r_i\in X\wedge r_i(a)=C,\ 1\leqslant i\leqslant|U|\}$

其中，$r_i(a)$为某一属性值，C 为要查询的属性值。

证明　设精确查询的结果为 A，而非精确查询的结果为 B，

对于任意的元素 $x \in A$，其等价类为$[x]$，而 A 为以等价关系为基础进行查询的结果，因此 A 是一些等价类的集合，显然 $x \in [x]$；而 B 为以共同关系为基础进行查询的结果，因此 B 为共同类的集合，设 x 在 B 中的共同类为$[x]_\theta$，其中θ为共同关系，显然等价类为共同类的子集，因此对于任意的 $x \in [x]$，必有 $x \in [x]_\theta$，亦即对于任意的 $x \in A$，必有 $x \in B$，因此 A 为最小集合，A 可以表示为：$A = \overline{R}A = \underline{R}A = \{r_i | r_i \in R \wedge r_i \in A \wedge r_i(a) = C, 1 \leqslant i \leqslant |U|\}$，其中：$r_i(a)$为某一属性值，$C$ 为要查询的属性值。定理得证。

表 3-1　子域

ID	COUNTRY	FEATURE
U123	US	{MARSH，LAKE}
U124	US	MARSH
U125	USA	{MARSH，PASTURE，RIVER}
U126	US	{FOREST，RIVER}
U147	US	{SAND，ROAD，URBAN}
U157	{US，MEXICO}	{SAND，ROAD}
M007	MEXICO	{SAND，ROAD}
M008	MEXICO	BEACH
M009	MEXICO	SAND
CO39	BELIZE	JUNGLE
CO40	{BELIZE，INT}	{JUNGLE，COAST，SEA}

表 3-2　子表 1

ID	COUNTRTY
U123	US
U124	US
U125	USA
U126	US
U147	US

3.4.2　粗糙完全查询

所谓对粗糙关系数据库的完全查询，是一种基于与查询关键字所有的可能匹配的数据查询。

进行粗糙完全查询，实际上要查询的是表中某一属性值或子解释的共同类，即所有等于及包含该属性值或子解释的元组，其方法为：

（1）对于要查询的数据，分析其共同类（含语义相同的属性值），然后根据所得到的共同类将 RRDB 分解为 RRDB 的一个子表，该子表中要查询的字段值及检索字段全部保留。

（2）分析得到的子表，若其为单值分解，则利用 SQL 实现数据查询，若其为多值分解，则利用粗糙关系操作（如投影）完成数据查询。

仍以表 3-1 为例，查询所有国家为 US 的所有 ID。根据完全查询方法，粗糙完全查询结果为{U123，U124，U125，U126，U147，U157}。从结果看，精确查询相当于下近似{U123，U124，U125，U126，U147}，而粗糙完全查询则等于上近似{U123，U124，U125，U126，U147，U157}，U157 为边界域元素。

定理 3.2　粗糙完全查询的结果是满足查询条件的最大集合，它可以表示为：$\overline{R}\ X=\{r_i|\ \exists\ i(r_i \in R) \wedge |r_i(a)| \geqslant 1 \wedge \Gamma_j(r_i 9a)\}=C$，$1\leqslant i\leqslant|U|$，$1\leqslant j\leqslant k$}。

其中，X 为结果集，k 为某个属性值的最大子解释数，$r_i(a)$为元组 r_i 基于属性 a 的属性值，$\Gamma_j(r_i(a))$为属性值 $r_i(a)$的一个子解释值，C 为某个要查询的属性值。

证明　设粗糙完全查询的结果为 A，而非粗糙完全查询的结果为 B，对于任意的元素 $x\in A$，已知 A 是以共同关系为基础进行查询的结果，因此 A 为共同类的集合，设 x 在 A 中的共同类为$[x]_0$，

其中θ为共同关系，显然等价类为共同类的子集，而 B 为等价类的集合，因此对于任意的 $x\in B$，必有 $x\in[x]_\theta$，亦即对于任意的 $x\in B$，必有 $x\in A$，因此 A 为包含所有查结果元素的集合，A 可以表示为：$A=\{r_i|\exists i(r_i\in R)\wedge|r_i(a)|\geqslant 1\ \Gamma_j(r_i(a))=C$，$1\leqslant i\leqslant|U|$，$1\leqslant j\leqslant k\}$。本定理得证。

3.4.3 粗糙组合查询

在粗糙关系数据库中，把几个查询条件结合起来进行数据检索，这种查询称为粗糙组合查询。

粗糙组合查询可分为粗糙精确组合查询和粗糙完全组合查询。粗糙精确组合查询是精确地查询符合组合条件的记录，它与精确查询类似。粗糙完全组合查询是查询所有符合组合条件的可能的记录，其方法为仍以表 3-1 为例查询所有 FEATURE 为{SAND，ROAD}的所有 ID，它的意思是同时具有 SAND 和 ROAD 结构的记录，可以表示为：

$\Pi_{1,3}$（$\sigma_{\text{FEATURE= “SAND”}\wedge\text{FEATURE= “ROAD”}}$（SUBREGIONS））。

BEACH 与 SAND 的语义是相似的，它们属于同一个等价类，因此从表 3-1 提取 1，3 列形成表 3-3 和表 3-4（方法同前）。

表 3-3　子表 2

ID	FEATURE
U147	SAND
U157	SAND
M007	SAND
M008	BEACH
M009	SAND

表 3-4　子表 3

ID	FEATURE
U147	ROAD
U157	ROAD
M007	ROAD

然后利用 SQL 语句实现查询，设对表 3-3 和表 3-4 的查询结果分别为 P，Q，则 P={U147，U157，M007，M008，M009}，Q={U147，U157，M007}，根据 Beaubouef • Theresa 的定义，两个 Rough 关系 P、Q 是并兼容的，因此最后的结果 $M=P\cap Q$={U147，U157，M007}。

3.4.4　算法描述

下面给出对于 RRDB 进行 Rough 数据查询的算法描述：

（1）Input S，K，A_i；//S 为一 RRDB，A_i 为要查询的属性，K 为查询种类；

（2）if k=0 then；

（3）input　Value1；

（4）求[Value1]；

（5）根据[Value1]及 A_i 对 S 进行分解得到 S'；

（6）if S'为单值分解 then；

（7）SELECT A_i FROM S′ WHERE $A_j\in$[Value1]；

（8）if S'为多值分解　then；

（9）利用高级语言完成字段值提取，goto（29）；

（10）if k=1 then；

（11）input　Value1；

（12）求$[\text{Value1}]_\theta$；//其中θ为共同关系；

（13）根据$[\text{Value1}]_\theta$及 A_i 对 S 进行分解得到 S'；

（14）if S'为单值分解 then；

（15）SELECT A_i FROM S′ WHERE $A_j \in [\text{Value 1}]_\theta$;

（16）if S'为多值分解 then；

（17）利用高级语言完成字段值提取，goto（29）；

（18）if k=2 then；

（19）input Value1，Value2…Value$k$；

（20）求$[\text{Value1}]_\theta$；$[\text{Value2}]_\theta \cdots [\text{Value}k]_\theta$；//其中θ为共同关系；

（21）根据$[\text{Value1}]_\theta$；$[\text{Value2}]_\theta \cdots [\text{Value}k]_\theta$及 A_i 对 S 进行分解得到 S_1，S_2，$\cdots S_K$；

（22）for I=1 to k；

（23）if $S_{[I]}$为单值分解 then；

（24）SELECT A_i FROM $S_{[I]}$ WHERE $A_j \in [\text{Value1}]$；

（25）进行粗糙关系操作（∩、∪）等；

（26）if $S_{[I]}$为多值分解 then；

（27）利用高级语言完成字段值提取；

（28）进行粗糙关系操作（∩、∪）等；

（29）输出查询结果，以表的形式输出。

可以看到在该算法描述中，使用了对 RRDB 求其等价类及共同类的方法，并且对于查询结果利用原 RRDB 的分解及相应的粗糙集来表示，这种处理从根本上节省了系统的存储空间，从而提高了系统的查询性能，因此使算法的查询得到了优化，该算法的时间复杂度为 $O(|S|^2)$。

3.4.5 小结

本章举例中的 ID 是该 RRDB 的关键字，也可以查询非关键字，这样的粗糙完全组合查询会更复杂些，但其原理是类似的，比如查询 FEATURE 为“SAND”及“ROAD”的 COUNTRY，这时取的分解为多值分解，即 FEATURE 对应的 COUNTRY 字段的属性值取其本身，而不取其子解释，FEATURE 取其子解释，

对这样形成的两个表再进行粗糙关系操作。

本章讨论的多值的子解释间是“或”的情况，把它分为精确查询、粗糙完全查询、粗糙精确组合查询和粗糙完全组合查询四种情况；若多值的子解释间是“与”的情况，也可分这四种情况，这时它的精确查询、粗糙完全查询、粗糙精确组合查询结果与“或”的粗糙完全查询结果相同，而“与”的粗糙完全组合查询则类似“或”的粗糙完全组合查询。

根据等价关系及共同关系原理、投影原理、分解原理本文给出了几种 RRDB 的数据查询，这些查询的基本原理可以归纳为：

（1）若查询的是 RRDB 的关键字或要查询的某列属性值为单值的情形，采用以下步骤：分析该关键字或属性值的等价类或共同类（基于等价关系、共同关系），利用投影原理提取该 RRDB 的单值分解后用 SQL 语句进行检索，并用粗糙关系操作完成最后处理。这里，RRDB 的查询结果实际上是一个粗糙集，它可以用相应的上、下近似来表示。

（2）若查询的是 RRDB 的非关键字或要查询的列属性值为多值的情形，采用以下步骤：分析非关键字的等价类或共同类，利用投影原理提取该 RRDB 的多值分解，最后用高级语言及粗糙关系操作完成最后处理。

3.5　RRDB与NIS的关系研究

RRDB 与 NIS 虽然在表示形式上是相同的，但是它们仍然存在着不同，具体表现在以下几方面：

（1）是否允许冗余。同经典的关系数据库一样，RRDB 也不允许粗糙关系之中出现冗余，也就是说，冗余的出现对于 RRDB 是没有意义的；而 NIS 则不同，在一个 NIS 中，是允许出现相同元组或冗余的，例如，表 3-6 所示的 NIS 是有意义的，其中，对象 1，3 的属性值相同。

表 3-5　不确定信息系统 1

OB	A	B	C
1	3	1	1
2	5	{2，4}	2
3	{1，4，5}	5	4
4	4	5	2
5	3	5	2
6	4	5	{1，3}
7	5	4	1
8	1	{1，3，4}	1

表 3-6　不确定信息系统 2

OB	A	B	C
1	{3}	{1，3，4}	{2}
2	{1}	{1，2}	{4}
3	{3}	{1，3，4}	{2}
4	{2}	{1，2}	{2，3}

（2）如同信息系统是关系数据库中关系的泛化一样，NIS 是 RRDB 的泛化，在 NIS 中，某个属性值（子集）的各元素之间的关系更多地为“或”，如表 3-5 中，{2，3}中 2 与 3 的关系第是“或”。而 RRDB 中，某个属性值（子集）的各元素之间的关系则既有“与”也有“或”，例如，表 3-1 的 ID 为 M007 的记录中，COUNTRY=“MEXICO”的 FEATURE 为既有“SAND”又有“ROAD”。

（3）RRDB 与 NIS 研究的问题领域不同。NIS 研究的问题主要集中在 The definability of a set in NISs，The consistency of an object，Data dependency in NISs，Rules in NISs，Reduction of attributes in NISs 及以上各问题的算法研究；RRDB 产生的主要动机是数据库的粗糙查询，其研究的主要问题则是粗糙关系运算、

数据库元组的可定义性、粗糙函数依赖、RRDB 的范式理论和粗糙数据查询等。

3.6　RRDB属性值的粗集表示

按照 T.Y.Lin 教授的观点，每个属性值是一个粒、一个等价类，当然也是一个对象集合，每个属性是一个二元关系，在粗糙集中，该二元关系为等价关系，把这个原理推广到粗糙关系数据库，来研究其粗糙集表示。

命题 3.1[121]　对于任一 RRDB 的粗糙关系来说，设 $t[A_j]$为其任一属性值，$\{X_1, X_2, \cdots, X_n\}$为其对象集，即其元组的集合，若 $t[A_j]\in X_i \wedge |t[A_j]|=1$，则 $X_i \subseteq \overline{R}_{tA_j}$，若 $t[A_j]\in X_i \wedge |t[A_j]|>1$，则 $X_i \subseteq \overline{R}_{tA_j}$，其中 $1\leqslant i\leqslant n$，$1\leqslant j\leqslant m$，$|\bullet|$表示 $t[A_j]$子解释的个数。

从表 3-1 可以看出，属性 ID 所对应的属性值均是唯一的，因此着重研究属性 COUNTRY 和 FEATURE，根据上述命题有：

$\underline{R}_{\mathrm{US}}=\{x_1, x_2, x_3, x_4, x_5\}$，$\overline{R}_{\mathrm{US}}=\{x_1, x_2, x_3, x_4, x_5, x_6\}$，

$\underline{R}_{\mathrm{MEXICO}}=\{x_7, x_8, x_9\}$，$\overline{R}_{\mathrm{MEXICO}}=\{x_6, x_7, x_8, x_9\}$，

$\underline{R}_{\mathrm{BELIZE}}=\{x_{10}\}$，$\overline{R}_{\mathrm{BELIZE}}=\{x_{10}, x_{11}\}$，

$\underline{R}_{\mathrm{MARSH}}=\{x_2\}$，$\overline{R}_{\mathrm{MARSH}}=\{x_1, x_2, x_3\}$，

$\underline{R}_{\mathrm{LAKE}}=\phi$，$\overline{R}_{\mathrm{LAKE}}=\{x_1\}$，

$\underline{R}_{\mathrm{PASTURE}}=\phi$，$\overline{R}_{\mathrm{PASTURE}}=\{x_3\}$，

$\underline{R}_{\mathrm{RIVER}}=\phi$，$\overline{R}_{\mathrm{RIVER}}=\{x_3, x_4\}$，

$\underline{R}_{\mathrm{FOREST}}=\phi$，$\overline{R}_{\mathrm{FOREST}}=\{x_4\}$，

$\underline{R}_{\mathrm{SAND}}=\{x_9\}$，$\overline{R}_{\mathrm{SAND}}=\{x_5, x_6, x_7, x_9\}$，

$\underline{R}_{\mathrm{ROAD}}=\phi$，$\overline{R}_{\mathrm{ROAD}}=\{x_5, x_6, x_7\}$，

$\underline{R}_{\mathrm{URBAN}}=\phi$，$\overline{R}_{\mathrm{URBAN}}=\{x_5\}$，

$\underline{R}_{\mathrm{BEACH}}=\{x_8\}$，$\overline{R}_{\mathrm{BEACH}}=\{x_8, x_{11}\}$，

$\underline{R}_{\text{JUNGLE}}=\{x_{10}\}$，$\overline{R}_{\text{JUNGLE}}=\{x_{10}, x_{11}\}$，

$\underline{R}_{\text{URBAN}}=\phi$，$\overline{R}_{\text{URBAN}}=\{x_{11}\}$。

分析上面各属性值的情况。可以看出，与经典关系数据库不同，按照粒计算的观点，经典关系 r 的属性值 $t[A_i]$ 满足 $\underline{R}_{t_{Ai}}=\overline{R}_{t_{Ai}}=[t[A_i]]$，其中 $[t[A_i]]$ 为属性值的等价类，说明其属性值可以看作是由一个等价类集合构成的粒，而在 RRDB 中，一般地，$\underline{R}_{t_{Ai}} \neq \overline{R}_{t_{Ai}}$，某个属性值 $t[A_i]$ 是由二元组（$\underline{R}_{t_{Ai}}$，$\overline{R}_{t_{Ai}}$）形式化表示的。

命题 3.2 在任一 RRDB 的粗糙关系中，对任意属性值 $t[A_i]$ 精确查询的结果为其下近似 $\underline{R}_{t_{Ai}}$，而对其粗糙查询的结果为其上近似 $\overline{R}_{t_{Ai}}$。

上述命题符合粗糙集理论中的下近似定义给出了被查询文档或相关查询的可定义的特性，而上近似则给出了相关文档或查询的可能被描述的特性。例如 $\underline{R}_{\underline{US}}=\{x_1, x_2, x_3, x_4, x_5\}$，为对 *US* 进行精确查询的结果，而 $\overline{R}_{\underline{US}}=\{x_1, x_2, x_3, x_4, x_5, x_6\}$，为对 US 进行粗糙查询的结果。

上述每个属性值为一个等价颗粒，它们对应的位表示为：

$\text{BIT}(\underline{R}_{\text{US}})=11111000000$，$\text{BIT}(\overline{R}_{\text{US}})=11111100000$，

$\text{BIT}(\underline{R}_{\text{MEXICO}})=00000011100$，$\text{BIT}(\overline{R}_{\text{MEXICO}})=00000111100$，

$\text{BIT}(\underline{R}_{\text{BELIZE}})=00000000010$，$\text{BIT}(\overline{R}_{\text{BELIZE}})=00000000011$，

$\text{BIT}(\underline{R}_{\text{MARSH}})=01000000000$，$\text{BIT}(\overline{R}_{\text{MARSH}})=11100000000$，

$\text{BIT}(\underline{R}_{\text{LAKE}})=00000000000$，$\text{BIT}(\overline{R}_{\text{LAKE}})=10000000000$，

$\text{BIT}(\underline{R}_{\text{PASTURE}})=00000000000$，$\text{BIT}(\overline{R}_{\text{PASTURE}})=00100000000$，

$\text{BIT}(\underline{R}_{\text{RIVER}})=00000000000$，$\text{BIT}(\overline{R}_{\text{RIVER}})=00110000000$，

$\text{BIT}(\underline{R}_{\text{FOREST}})=00000000000$，$\text{BIT}(\overline{R}_{\text{FOREST}})=00010000000$，

$\text{BIT}(\underline{R}_{\text{SAND}})=00000000100$，$\text{BIT}(\overline{R}_{\text{SAND}})=00001110100$，

$\text{BIT}(\underline{R}_{\text{ROAD}})=00000000000$，$\text{BIT}(\overline{R}_{\text{ROAD}})=00001110000$，

BIT($\underline{R}_{\text{URBAN}}$)=00000000000，BIT($\overline{R}_{\text{URBAN}}$)=00001000000，

BIT($\underline{R}_{\text{BEACH}}$)=00000001000，BIT($\overline{R}_{\text{BEACH}}$)=00000001001，

BIT($\underline{R}_{\text{JUNGLE}}$)=00000000010，BIT($\overline{R}_{\text{JUNGLE}}$)=00000000011，

BIT($\underline{R}_{\text{URBAN}}$)=00000000000，BIT($\overline{R}_{\text{URBAN}}$)=00000000001。

利用粗糙集理论与位方式对 RRDB 的属性值进行粒化表示后，便可以对它进行方便地查询，这里把基于位方式的查询分为：

（1）确定的查询。即精确查询，也就是利用粗糙集理论的下近似所得到的查询结果。

如查询“COUNTRY 为 US 的对象”，利用前面对于 US 的下近似的结果，即 BIT($\underline{R}_{\text{US}}$) =11111000000，结果为{$x_1$，$x_2$，$x_3$，$x_4$，$x_5$}；若查询“COUNTRY 为 US 且 FEATURE 为 MARSH 的对象”，则取其下近似的位表示的交，即 BIT($\underline{R}_{\text{US}}$)∩BIT($\underline{R}_{\text{MARSH}}$)= 11111000000 ∩ 01000000000 =01000000000，结果为{X_2}。

以上查询基于粗糙集原理 $\underline{R}(X \cap Y)= \underline{R}(X) \cap \underline{R}(Y)$。

（2）可能的查询。即粗糙查询，亦即利用粗糙集理论的上近似所得到的查询结果，比如：查询“COUNTRY 为 US 的对象”，利用前面对于 US 所求的上近似的结果，即 BIT($\overline{R}_{\text{US}}$)=111111000000，其结果为{x_1，x_2，x_3，x_4，x_5，x_6}；若查询“COUNTRY 为 US 且 FEATURE 为 MARSH 的对象”，则取其上近似位表示的交，即 BIT($\overline{R}_{\text{US}}$)∩BIT($\overline{R}_{\text{MARSH}}$)=111111000000∩ 11100000000 =11100000 000，其结果为对象集{x_1，x_2，x_3}。

以上查询基于粗糙集原理 $\overline{R}(X \cap Y) \subseteq \overline{R}(X) \cap \overline{R}(Y)$。

本章首先分解粗糙关系数据库，然后再对分解进行查询。进行分解的主要意图有两个：一是利用分解保留查询的中间结果，这样可以辅助其他相关工作；二是粗糙关系数据库是不能用 SQL 的，若进行分解后为单值分解，则可以对该分解直接利用 SQL 来完成查询，这样可以提高查询效率。若查询中无其他特殊的要

求，对本章方法的改进可以按照本文分析的粗糙关系数据库的粗糙查询原理，省去进行粗糙关系数据库分解这步，直接对粗糙关系数据库利用粗糙关系原理与高级语言进行查询，这样可以达到进一步的优化。

第 4 章　粗糙函数依赖及其推理机制的研究

虚心涵泳，切己体察
——曾国藩《字谕纪泽儿》

4.1　引言

关系数据库是以关系模型为基础的，它利用关系描述现实世界。数据依赖是通过一个关系中属性值的相等与否体现出来的数据间的相互关系，是现实世界属性间相互联系的抽象，是语义的体现，有许多类型的数据依赖，其中最重要的是函数依赖和多值依赖。

函数依赖是关系数据库最重要的概念之一。数据库的属性间往往存在一定的依赖关系，而最基本的依赖关系是函数依赖，所谓函数依赖是指一个关系中一个或一组属性的值能够决定其他属性的值。函数依赖一般用于数据库的逻辑设计，用以表示完整性约束。

对于经典的函数依赖来说，其推理规则和附加的推理规则早已被证明，模糊（Fuzzy）函数依赖的推理规则和附加的推理规则也已由一些相关文献给出[134]，Theresa Beaubouef 在文献[42]中给出了粗糙函数依赖的基本定义，但并没有给出粗糙函数依赖关于 Armstrong 公理的详细证明，也没有涉及附加的推理规则（合并、分解、伪传递规则）。

本章中在分析粗糙函数依赖基本定义的基础上，提出了冗余

因子的概念，用于研究粗糙函数依赖的推理规则。具体地，在引入粗糙函数依赖的基础上提出了先行上、下冗余因子及结果上、下冗余因子的概念，讨论了粗糙函数依赖的性质、推理规则和附加的推理规则，并且分析了粗糙函数依赖与函数依赖、Fuzzy 函数依赖三者之间的关系。

4.2 函数依赖与模糊函数依赖

定义 4.1 设 $R=\{A_1, A_2, \cdots, A_n\}$为具有 n 个属性的关系模式，X，$Y \subseteq A$，X，Y 之间的函数依赖记为 $X \rightarrow Y$，其依赖性描述为当 $t_i[X]=t_j[X]$时，必有 $t_i[Y]=t_j[Y]$，其中 t_i，t_j 为数据库的两个元组，称 X 函数决定 Y，或 Y 函数依赖于 X。

对于函数依赖，需要说明以下几点：

（1）函数依赖不是指关系模式 R 的某个或某些关系实例满足的约束条件，而是指 R 的所有关系实例均要满足的约束条件。

（2）函数依赖和别的数据之间的依赖性一样，是语义范畴的概念，只能根据数据的语义来确定函数依赖。

（3）数据库设计者可以对现实世界做强制的规定，即指定相应的函数依赖。

定义 4.2[124] 对于满足一组函数依赖 F 的关系模式 $R<U, F>$，其任意一个关系 r，若函数依赖 $X \rightarrow Y$ 都成立，（即 r 中任意两元组 t，s，若 $t[X]=s[X]$，则 $t[Y]=s[Y]$），则称 F 逻辑蕴含 $X \rightarrow Y$。

为了从一组函数依赖求得蕴含的函数依赖，如已知函数依赖集 F，要问 $X \rightarrow Y$ 是否为 F 所蕴含，就需要一套推理规则，这组推理规则是 1974 年首先由 Armstrong 提出来的。

Armstrong **公理系统**[124] 设 U 为属性集总体，F 是 U 上的一组函数依赖，于是有关系模式 $R<U, F>$，对 $R<U, F>$来说有以下的推理规则：

（1）自反律（Reflexivity）：若 $Y \subseteq X \subseteq U$，则 $X \rightarrow Y$ 为 F 所蕴含。

（2）增广律（Augmentation）：若 $X \rightarrow Y$ 为 F 所蕴含，且 $Z \subseteq U$，则 $XZ \rightarrow YZ$ 为 F 所蕴含。

（3）传递律（Transitivity）：若 $X \rightarrow Y$ 及 $Y \rightarrow Z$ 为 F 所蕴含，则 $X \rightarrow Z$ 为 F 所蕴含。

Armstrong 公理的有效性指由 F 出发根据 Armstrong 公理推导出来的每一个函数依赖一定在 F^+中，F^+为 F 的闭包，即 F 所蕴含的函数依赖的全体。

Armstrong 公理的完备性指 F^+中的每一个函数依赖，必定可以由 F 出发根据 Armstrong 公理推导出来。

引理 4.1[124]　Armstrong 推理规则是正确的。

引理 4.2[124]　Armstrong 公理系统是有效的和完备的。

定义 4.3　关系模式 R（A_1，…，A_n）上的模糊关系是 $\mathrm{dom}(A_1) \times \cdots \times \mathrm{dom}(A_n)$的模糊子集。

把函数依赖的概念推广应用到模糊关系数据库中，就形成了模糊函数依赖，模糊函数依赖可以灵活地表示模糊关系中属性值间的依赖关系，比如“某人的薪水依赖于其工作中的职位与经验”等。普通函数依赖的模糊扩展应当满足下列条件：

（1）普通函数依赖必须是模糊函数依赖的特例，也就是说，如果一个关系满足模糊函数依赖，那么它必须满足经典的函数依赖。

（2）不相似的先行值不影响函数的依赖性。

（3）对于一个实例来说，每个属性有它自己的特性，一个属性对于另外一个属性，其相似的程度一般是不同的。

Buckles 与 Petry 把模糊信息融合到关系数据库中形成了模糊关系数据库，它是经典关系数据库的泛化[135]。

模糊函数依赖一般是基于相似关系的，它利用模糊隶属度及属性值之间的相似关系来确定相应的函数依赖。

定义 4.4[136] 设 D 为论域，其上的相似关系是一个映射 R：$D\times D\to[0,1]$，用 $R(x,y)$ 来表示 x 与 y 的相似度，相似关系满足自反性、对称性，即：

$R(x,x)=1$ 自反性

$R(x,y)=R(y,x)$ 对称性

定义 4.5 设 D 为一论域，R 为定义在 $D\times D$ 上的相似关系，称 $x,y\in D$ 为 α 级相似，当且仅当 $R(x,y)\geqslant\alpha$。我们也把它称为 α 级冗余，记为 $x\,\square_{\alpha}\,y$。

定义 4.6[42] 对于模糊关系数据库 $R=\{A_1,A_2,\cdots,A_n\}$，设其任意的两个元组是 α 级冗余的（记为 $\square_{\alpha}$），t_1、t_2 为 R 的任意两个元组，X，$Y\subseteq R$ 为 R 的属性子集，X，Y 之间的 Fuzzy 函数依赖 $X\xrightarrow[(\alpha X,\alpha Y)]{}Y$ 成立当且仅当 $t_1(X)\,\square_{\alpha X}\,t_2(X)$ 时，有 $t_1(Y)\,\square_{\alpha Y}\,t_2(Y)$，其中 $\square_{\alpha X}$、$\square_{\alpha Y}$ 均为 α 级冗余。

本书中，把模糊函数依赖记为：$X\xrightarrow{F}Y$。

引理 4.3[136] 经典的函数依赖（FD）满足模糊函数依赖（FFD）的定义。

模糊函数依赖满足以下推理规则[137]：

FA1(Reflexivity):if $Y\subseteq X\subseteq U$ then $X\xrightarrow{F}Y$。

FA2(Augmentation):if $X\xrightarrow{F}Y$ and $Z\subseteq U$，then $XZ\xrightarrow{F}YZ$。

FA3(Transitivity):if $X\xrightarrow{F}Y$ and $Y\xrightarrow{F}Z$，then $X\xrightarrow{F}Z$。

FA9 (Union):if $X\xrightarrow{F}Y$ and $X\xrightarrow{F}Z$，then $X\xrightarrow{F}YZ$。

FA10 (Decomposition):if $X\xrightarrow{F}YZ$，then $X\xrightarrow{F}Y$ and $X\xrightarrow{F}Z$。

FA11 (Pseudotransitivity):if $X\xrightarrow{F}Y$ and $YW\xrightarrow{F}Z$，then $XW\xrightarrow{F}Z$。

引理 4.4[136] 推理规则 FA1-FA3 及 FA9-FA11 是有效的。

注意一般来说模糊函数依赖的推理规则是不完备的。

4.3　粗糙函数依赖与冗余因子

把由粗糙关系数据库模型支持的数据库称为粗糙关系数据库（Rough Relational Database，RRDB），下面给出与 RRDB 有关的函数依赖的相关概念，并对其中关键的概念加以分析。

定义 4.7[42]　（元组冗余）设 R 为任一粗糙关系数据库模式，r 为其任意粗糙关系，$t_i=(d_{x1}, d_{x2}, \cdots, d_{xm})$，$t_j=(d_{y1}, d_{y2}, \cdots, d_{ym})$为 r 的任意两个子元组（子元组是元组的子集），对所有的 $j=1, 2, \cdots, m$，若$[d_{xj}]=[d_{yj}]$，则称 t_i，t_j 是元组冗余（Redundant）的，其中$[d]$表示包含 d 的等价类。

定义 4.8[42]　（粗糙冗余）设 R 为任一粗糙关系数据库模式，r 为其任意粗糙关系，设 $t_i=(d_{x1}, d_{x2}, \cdots, d_{xm})$，$t_j=(d_{y1}, d_{y2}, \cdots, d_{ym})$为 r 的任意两个子元组，对所有的 $j=1, 2, \cdots, m$，当存在 p、q 使得$[p]\subset[d_{xj}]$，$[q]\subset[d_{yj}]$时，有$[p]=[q]$成立，则称 t_i，t_j 是粗糙冗余（Rough Redundant）的。

定义 4.9[42]　设 R 为任一粗糙关系数据库模式，T 为其任意粗糙关系，设 X，Y 为粗糙关系模式 R 的属性子集，粗糙函数依赖（RFD）$X\rightarrow Y$ 对于一个粗糙关系模式 R 的所有实例 T 都成立，当满足：

（1）对于任意两个元组 t，$t'\in \underline{R}\,T$ 时，

Redundant($t(X)$，$t'(X)$)→Redundant($t(Y)$，$t'(Y)$)成立。

（2）对于任意两个元组 s，$s'\in \overline{R}\,T$ 时，下式成立：

Rough-redundant($s(X)$, $s'(X)$)→Rough-redundant($s(Y)$, $s'(Y)$)。

其中 $\underline{R}\,T$、$\overline{R}\,T$ 为 Rough 关系模式 R 的所有实例 T 的下近似与上近似，对于实例 T 的上、下近似的定义，文献[42]并没有给出，只是笼统地使用。这里把它做如下理解：若 R 为数据库原始关系，则 $\overline{R}\,T=\underline{R}\,T=R$；若 R 为对 RRDB 进行粗糙关系操作的集

合，则 T 的上、下近似可定义为：

$\overline{R}\,T=\{r_i|\exists\, i(r_i\in R)\wedge|r_i（a_j）|\geqslant 1\wedge r_i\cap T\neq\phi$，$1\leqslant i\leqslant n$，$1\leqslant j\leqslant m\}$

$\underline{R}\,T=\{r_i|r_i\in R\wedge|r_i（a_j）|=1\wedge r_i\in T$，$1\leqslant i\leqslant n$，$1\leqslant j\leqslant m\}$。

这里 T 为元组的集合，r_i 为它的一个元组，$r_i（a_j）$ 为与关系操作相关的元组 r_i 的一个属性值，$|r_i（a_j）|$为组成属性值 $r_i（a_j）$ 的原子值的个数，n 为 T 的元组的个数，a_j 为第 j 个属性，j 为属性数。

从上面的定义可以看出，元组冗余是用于定义属于下近似的元组的函数依赖的，Rough 冗余是用于定义属于上近似的元组的函数依赖的。同时，元组冗余是基于等价关系的，而粗糙冗余是基于相似关系的，因此，粗糙函数依赖是以等价关系和相似关系为基础的。当 $\overline{R}\,T=\underline{R}\,T$ 时，粗糙冗余即为元组冗余，粗糙函数依赖则变为普通的函数依赖。

在表 4-1 的粗糙关系数据库 S[42]中，有粗糙函数依赖 ID→FEATURE 存在，因为对于元组{U123，US，{MARSH，LAKE}}与{{U123， U124}， US， {MARSH， LAKE， GRASS}}来说，{U123， {MARSH， LAKE}}与{{U123， U124}， {MARSH，LAKE，GRASS}}为其子元组，{U123}与{U123，U124}是粗糙冗余的，{MARSH，LAKE}与{MARSH，LAKE，GRASS}也是粗糙冗余的，而其余的 ID 均不同，因此有 RFD：ID→FEATURE 成立。

表 4-1　地理信息中的粗糙关系

ID	COUNTRY	FEATURE
U123	US	{MARSH，LAKE}
U125	USA	{MARSH，PASTURE，RIVER}
U130	US	{FOTEST，GRASS}
U135	US	FOREST
U136	US	WOODS
{U123，U124}	US	{MARSH，LAKE，GRASS}

定义 4.10　（冗余因子）称定义 4.9 的 Redundant($t(X)$，$t'(X)$)中 $t(X)$与 $t'(X)$的相似度为先行下冗余因子，用 α 表示，称 Redundant($t(Y)$，$t'(Y)$)中 $t(Y)$与 $t'(Y)$的相似度为结果下冗余因子，用 β 表示，则定义 4.9 之(1)可记为 $X_\alpha \rightarrow Y_\beta$，称表达式 Rough-redundant($s(X)$，$s'(X)$)中 $s(X)$与 $s'(X)$的相似度为先行上冗余因子，用 α' 表示，称表达式 Rough-redundant($s(Y)$，$s'(Y)$)中 $s(Y)$与 $s'(Y)$的相似度为结果上冗余因子，用 β' 表示，定义 4.9 之（2）可记为 $X'_\alpha \xrightarrow{R} Y'_\beta$，其中：

$$\alpha = \frac{card(t(X) \cap t'(X))}{card(t(X) \cup t'(X))},\quad \beta = \frac{card(t(Y) \cap t'(Y))}{card(t(Y) \cup t'(Y))} \qquad \alpha、\beta \in [0，1]$$

其中 $card(\ )$表示基数，α'，β' 的定义与 α，β 类似。

上例中，{U123}与{U123，U124}是 Rough 冗余的，其 α' =1/2=0.5，{MARSH，LAKE}与{MARSH，LAKE，GRASS}也是粗糙冗余的，因此 β'=2/3。

用 $X \xrightarrow{R} Y$ 来表示粗糙函数依赖，或用 $X_\alpha \rightarrow Y_\beta$ 及 $X'_\alpha \xrightarrow{R} Y_\beta$ 来表示。

4.4　Rough函数依赖的性质

性质 4.1　设 R 为任一粗糙关系数据库模式，r 为其任意粗糙关系，对于满足粗糙函数依赖 $X \xrightarrow{R} Y$ 的任意 r，它的任意两个元组 t=（d_{x1}，d_{x2}，…，d_{xm}），t'= (d_{y1}，d_{y2}，…，d_{ym})若是元组冗余的，则它们必是粗糙冗余的，若 t，t'是粗糙冗余的，它们未必是元组冗余的。

根据元组冗余和粗糙冗余的定义，任意两个元组若是元组冗余的，则其 α =1，其相似度或冗余度为 100%，那么它们显然是粗糙冗余的，但粗糙冗余的两个元组只有在 α'=1 且 β'=1 时才是元组冗余的。

性质 4.2 设 R 为任一粗糙关系数据库模式，r 为其任意粗糙关系，粗糙函数依赖 $X_\alpha \to Y_\beta$ 及 $X'_\alpha \xrightarrow{R} Y_{\beta'}$ 中，α，$\beta \in \{0, 1\}$，即其取值为 0 或 1；α'，$\beta' \in [0, 1]$，即 α'，β' 取 0 与 1 之间的任意值（含 0，1）。若 $\alpha=1$，则必有 $\beta=1$；但若 $\alpha'=1$，则 β' 的值未必为 1。

若 RRDB 满足粗糙函数依赖，由定义 4.9 之（1）可得任意两个元组间的 α 值取 1 或 0，α 值取 1，则 β 取 1；α 值取 0，则 β 不做要求；对于任意两个元组间的 α'，β' 来说，由定义 4.9 之（2）有若 $\alpha'=1$，则 β' 只要不为 0 即可。

性质 4.3 在粗糙函数依赖中，不相似的先行值不影响函数的依赖性，即若 α，α' 为 0，则 β，β' 可为任何值。

这个性质是由关系数据库中函数依赖本身的特点决定的。

从粗糙函数依赖的定义可以看出，粗糙函数依赖是普通函数依赖的泛化，普通函数依赖用于关系数据库中，而粗糙函数依赖用于多值（属性值可能是一集合）的粗糙关系数据库中，它同 Fuzzy 函数依赖一样，是普通函数依赖的泛化，普通的函数依赖是它的特例，因而有下列定理。

定理 4.1 普通的函数依赖满足粗糙函数依赖 $X_\alpha \to Y_\beta$，$X_{\alpha'} \xrightarrow{R} Y_{\beta'}$。

证明：设 t_i，t_j 为数据库的任意两个元组，X，Y 为数据库的属性，若存在函数依赖 $X \to Y$，意味着 $t_i[X]=t_j[X]$成立，则必有 $t_i[Y]=t_j[Y]$，则此时 α，β，α'，β' 的值均为 1，t_i，t_j 既是元组冗余（等价关系成立）的也是粗糙冗余的（相似关系成立），因此关系数据库是粗糙关系数据库的特例，而关系数据库的函数依赖满足粗糙函数依赖的定义，它是一种特殊的粗糙函数依赖。

该定理还可表述为：粗糙函数依赖与普通的函数依赖是一致的，定理 4.1 也可以称为一致性定理。

性质 4.4 在粗糙函数依赖中，α'，β' 的值越大，则粗糙函

数依赖的程度越高。

在用于知识发现的信息系统中，函数依赖对应于确定的决策规则，而部分依赖用于可能的决策规则，即默认规则，而粗糙关系数据库是非第一范式的，若把它转化为第一范式，即把粗糙关系数据库转变为关系数据库，则定义 4.9 中的（2）将转变为部分依赖；或者说 4.9（1）对应的元组反映的是信息系统中的确定性规则，而 4.9（2）中的元组对应的是默认规则。

4.5　粗糙函数依赖的推理规则与附加的推理规则

4.5.1　粗糙函数依赖的推理规则

设 U 为属性集总体，F 是 U 上的一组粗糙函数依赖，于是有粗糙关系模式 $R<U, F>$，对于 $R<U, F>$来说有以下粗糙函数依赖推理规则（Armstrong 公理）：

RFD1：自反律若 $Y \subseteq X \subseteq U$，则 $X \xrightarrow{R} Y$ 为 F 所蕴含。

RFD2：传递律若 $X \xrightarrow{R} Y$，$Y \xrightarrow{R} Z$ 为 F 所蕴含，则 $X \xrightarrow{R} Z$ 为 F 所蕴含。

RFD3：增广律若 $X \xrightarrow{R} Y$ 为 F 所蕴含，且 $Z \subseteq U$，则 $XZ \xrightarrow{R} YZ$ 为 F 所蕴含。

定理 4.2　公理 RFD1，RFD2，RFD3 是正确的。

证明：分别证明它们的正确性。

RFD1　设 t，t'为 RRDB 的任意两个元组，t、$t' \in \underline{R}\, T$，现有 $Y \subseteq X \subseteq U$ 成立，因而 $X \cap Y = Y$，若有 Redundant($t(X)$，$t'(X)$)，则按照元组冗余的定义，$t(X)=t'(X)$成立，属性 Y 为 X 的子集，因此 Y 满足 $t(Y)=t'(Y)$，故有 Redundant($t(Y)$，$t'(Y)$)，所以 Redundant($t(X)$，$t'(X)$) $\rightarrow$ Redundant ($t(Y)$，$t'(Y)$)成立；设 t、$t' \in \overline{R}\, T$，若有 Rough-redundant($t(X)$，$t'(X)$)，按粗糙冗余的定义有 $t(X) \cap t'(X) \neq \phi$，而 Y 为 X 的子集，因而 $t(Y) \cap t'(Y) \neq \phi$，Rough-redundant($t(Y)$，$t'(Y)$)

成立，即 Rough-redundant($t(X)$，$t'(X)$)$\rightarrow$Rough-redundant($t(Y)$, $t'(Y)$)成立，故 $X \xrightarrow{R} Y$ 为 F 所蕴含，RFD1 得证。

RFD2　设 R 为任一粗糙关系数据库模式，若其粗糙函数依赖 $X \xrightarrow{R} Y$，$Y \xrightarrow{R} Z$ 成立，对于任意两个元组 t，$t' \in \underline{R}\, T$，有 Redundant($t(X)$, $t'(X)$)$\rightarrow$Redundant($t(Y)$, $t'(Y)$), Redundant($t(Y)$, $t'(Y)$)$\rightarrow$Redundant($t(Z)$，$t'(Z)$)，即 $t(X)=t'(X)$成立，有 $t(Y)=t'(Y)$成立，而由 $t(Y)=t'(Y)$成立可以推出 $t(Z)=t'(Z)$成立，所以对于两个元组 t，t'来说，$t(X)=t'(X)$成立必有 $t(Z)=t'(Z)$成立，即

$$\text{Redundant}(t(X)，t'(X)) \rightarrow \text{Redundant}(t(Z)，t'(Z)) \quad (4.1)$$

同理可以证得，对于任意两个元组 t，$t' \in \overline{R}\, T$，有

$$\text{Rough-redundant}(t(X)，t'(X)) \rightarrow \text{Rough-redundant}(t(Z)，t'(Z)) \quad (4.2)$$

成立，由（4.1）、（4.2）$X \xrightarrow{R} Z$ 为 F 所蕴含，故传递性得证。

根据粗糙函数依赖的定义，它传递的是一种“弱相似关系”或者说是“对应的属性值相交不为空”，而不是传递的相似度（与模糊函数依赖不同）。

RFD3　设 R 为任一粗糙关系数据库模式，T 为其任意粗糙关系，设 X，Y，Z 为 T 的属性子集，设 t，t'为 RRDB 的任意两个元组，t，$t' \in \underline{R}\, T$，若 $X \xrightarrow{R} Y$，即 Redundant($t(X)$，$t'(X)$)$\rightarrow$redundant($t(Y)$，$t'(Y)$)，亦即 $t(X)=t'(X) \rightarrow t(Y)=t'(Y)$；若 $t(\text{XZ})=t'(\text{XZ})$，则有 $t(X)=t'(X)$，$t(Z)=t'(Z)$，因而对于两个元组 t，t'来说有 $t(\text{YZ})=t'(\text{YZ})$成立，故 Redundant($t(\text{XZ})$，$t'(\text{XZ})$)$\rightarrow$Redundant($t(\text{YZ})$，$t'(\text{YZ})$)为 F 所蕴含；同理，对于任意两个元组 t，$t' \in \overline{R}\, T$，有 Rough-redundant($t(\text{XZ})$，$t'(\text{XZ})$)$\rightarrow$Rough-redundant($t(\text{YZ})$，$t'(\text{YZ})$)，因此 $\text{XZ} \xrightarrow{R} \text{YZ}$ 为 F 所蕴含，RFD3 得证。

推论 4.1　粗糙函数依赖关于 Armstrong 公理是有效的。

4.5.2　粗糙函数依赖的附加推理规则

除 Armstrong 公理外，对于粗糙函数依赖，还有下面附加的

推理规则：

RFD4：合并规则若 $X \xrightarrow{R} Y$，$X \xrightarrow{R} Z$，则 $X \xrightarrow{R} \mathrm{YZ}$ 成立。

RFD5：分解规则若 $X \xrightarrow{R} Y$ 及 $Z \subseteq Y$，则 $X \xrightarrow{R} Z$ 成立。

RFD6：伪传递规则若 $X \xrightarrow{R} Y$，$\mathrm{WY} \xrightarrow{R} Z$，则 $\mathrm{XW} \xrightarrow{R} Z$ 成立。

定理 4.3　规则 RFD4，RFD5，RFD6 是正确的。

证明：分别证明它们的正确性。

RFD4　若粗糙函数依赖 $X \xrightarrow{R} Y$，$X \xrightarrow{R} Z$ 成立，对于任意两个元组 t，$t' \in \underline{R}\ T$，有 $\mathrm{Redundant}(t(X), t'(X)) \rightarrow \mathrm{Redundant}(t(Y), t'(\mathrm{Y}))$，$\mathrm{Redundant}(\mathrm{t(X)}, t'(\mathrm{X})) \rightarrow \mathrm{Redundant}(t(Z), t'(Z))$，即 $t(X)=t'(X) \rightarrow t(Y)=t'(Y)$，$t(X)=t'(X) \rightarrow t(Z)=t'(Z)$成立，则对于任意两个元组 t，t'来说 $t(\mathrm{YZ})=t'(\mathrm{YZ})$成立，即 $\mathrm{Redundant}(t(X), t'(X)) \rightarrow \mathrm{Redundant}(t(\mathrm{YZ}), t'(\mathrm{YZ}))$成立；设 t、$t' \in \overline{R}\ T$，有蕴含式 $\mathrm{Rough\text{-}redundant}(t(X), t'(X)) \rightarrow \mathrm{Rough\ redundant}(t(Y), t'(Y))$ 及 $\mathrm{Rough\text{-}redundant}(t(X), t'(X)) \rightarrow \mathrm{Rough\text{-}redundant}(t(Z), t'(Z))$成立，也就是意味着 $t(X) \cap t'(X) \neq \phi \rightarrow t(Y) \cap t'(Y) \neq \phi$，$t(X) \cap t'(X) \neq \phi \rightarrow t(Z) \cap t'(Z) \neq \phi$ 成立，则对于任意两个元组 t，t'来说 $t(\mathrm{YZ}) \cap t'(\mathrm{YZ}) \neq \phi$ 成立，即 $\mathrm{Rough\text{-}redundant}(t(X), t'(X)) \rightarrow \mathrm{Rough\text{-}redundant}(t(\mathrm{YZ}), t'(\mathrm{YZ}))$，因此 $X \xrightarrow{R} \mathrm{YZ}$ 为 F 所蕴含，该规则得证。

RFD5　若函数依赖 $X \xrightarrow{R} Y$ 成立，对于任意两个元组 t，$t' \in \underline{R}\ T$，有 $\mathrm{Redundant}(t(X), t'(X)) \rightarrow \mathrm{Redundant}(t(Y), t'(Y))$，即 $t(X)=t'(X) \rightarrow t(Y)= t'(Y)$；现有 $Z \subseteq Y$，设 $Y=Z \cup Z'$，则有 $t(Z)=t'(Z)$，$t(Z')=t'(Z')$，故有 $\mathrm{Redundant}\ (t(X), t'(X)) \rightarrow \mathrm{Redundant}(t(Z), t'(Z))$成立；同理可证对任意两个元组 t、$t' \in \overline{R}\ T$，$\mathrm{Rough\text{-}redundant}(t(X), t'(X)) \rightarrow \mathrm{Rough\text{-}redundant}\ (t(Z), t'(Z))$，因此 $X \xrightarrow{R} Z$ 为 F 所蕴含，RFD5 得证。

RFD6　若函数依赖 $X \xrightarrow{R} Y$ 成立，则对于任意两个元组

t，$t' \in \underline{R}\,T$，有蕴含式 Redundant($t(X)$，$t'(X)$) → Redundant($t(Y)$，$t'(Y)$)，即 $t(X)=t'(X) \to t(Y)=t'(Y)$；若 t(XW)=t'(XW)，则 $t(X)=t'(X)$，$t(W)=t'(W)$；现已知 WY $\xrightarrow{R}$ Z 成立，即 Redundant(t(WY)，t'(WY)) → Redundant($t(Z)$，$t'(Z)$)成立，亦即 t(WY)=t'(WY) → $t(Z)=t'(Z)$，即由 $t(Y)=t'(Y)$，$t(W)=t'(W)$可得 $t(Z)=t'(Z)$；由于有 $t(X)=t'(X) \to t(Y)=t'(Y)$，因此，若 t(XW)=t'(XW)成立，对于任意两个元组 t，$t't(Z)=t'(Z)$也成立，即 Redundant(t(XW)，t'(XW)) → Redundant($t(Z)$，$t'(Z)$)成立；同理可证对任意两个元组 t、$t' \in \overline{R}\,T$，Rough-Redundant(t(XW)，t'(XW)) → Rough-redundant($t(Z)$，$t'(Z)$)成立，所以 XW $\xrightarrow{R}$ Z 为 F 所蕴含，本规则得证。

推论 4.2 Rough 函数依赖关于 Armstrong 公理的附加推理规则是有效的。

定理 4.4 粗糙冗余关系是相似关系。

证明：根据定义，对于任意的两个子元组 t_1=（d_{x1}，d_{x2}，…，d_{xm}），t_2= (d_{y1}，d_{y2}，…，d_{ym})，t_1、t_2 是粗糙冗余的，则有[p]⊂[d_{xj}]，[q]⊂[d_{yj}]，[p]=[q]成立，j=1，2，…，m，对于任意的对应位置的属性值 d_{xi}、d_{Yi} 来说，有 R（d_{xi}，d_{xi}）=1，且有 R（d_{xi}，d_{yi}）= R（d_{yi}，d_{xi}），这是因为 $d_{xi} \cap d_{yi} = d_{yi} \cap d_{xi}$，满足相似关系的定义。因此粗糙冗余关系从本质上说是相似关系。

定理 4.5 RFD1～RFD3 是完备的。

像普通函数依赖一样，只需证明完备性的逆否命题，即若粗糙函数依赖 $X \xrightarrow{R} Y$ 不能由 F 从 Armstrong 公理推导出来，那么它必然不为 F 所蕴含。对于粗糙函数依赖 $X \xrightarrow{R} Y$ 来说，它由依托等价关系的 Redundant($t(X)$，$t'(X)$) → Redundant($t(Y)$，$t'(Y)$)和依托相似关系的 Rough-redundant($s(X)$，$s'(X)$) → Rough-Redundant($s(Y)$，$s'(Y)$)构成，这里的相似关系实际上是一种程度等价关系，只不过等价关系的相似度为 1，相似关系实际上减弱了等价性，即放松了对等价性的要求，因此，对于这里完备性的证明可以套用

Armstrong 公理系统完备性的证明过程（分三步完成）[90]，这里不再赘述了。

4.6　粗糙函数依赖与函数依赖、Fuzzy函数依赖的关系

通过比较函数依赖、Fuzzy 函数依赖和粗糙函数依赖，可以得出下面几点结论：

（1）它们的应用背景不同。传统的关系数据库使用的是普通的函数依赖，RRDB 是关系数据库的扩充。根据定义，粗糙函数依赖是用于支持 RRDB 的，而 Fuzzy 函数依赖则既可用于关系数据库，亦可用于 Fuzzy 关系数据库。

（2）粗糙函数依赖与 Fuzzy 函数依赖均是普通函数依赖的泛化。Fuzzy 函数依赖利用相似关系减弱等价性来泛化函数依赖，粗糙函数依赖利用不可分辨关系与相似关系来泛化函数依赖，函数依赖是粗糙函数依赖和 Fuzzy 函数依赖的特例，在特殊的情形下，粗糙函数依赖和 Fuzzy 函数依赖均可转化为普通的函数依赖。

（3）粗糙函数依赖与 Fuzzy 函数依赖同普通的函数依赖一样均满足 Armstrong 公理及其附加的推理规则（Fuzzy 函数依赖及普通函数依赖满足 Armstrong 公理及其推理规则已被证明）。

（4）粗糙函数依赖与 Fuzzy 函数依赖泛化函数依赖的形式不同：粗糙函数依赖利用上、下近似与谓词描述来定义，Fuzzy 函数依赖则利用相似关系等来扩充函数依赖，从对等价性的要求看，粗糙函数依赖比 Fuzzy 函数依赖要更弱一些，它并不要求 α'，β' 是同一数量级的，只要求具有函数依赖的属性对应的属性值包含相同的元素即可；而 Fuzzy 函数依赖则要求具有函数依赖的属性对应的属性值具有同等的相似度。

（5）像 Fuzzy 函数依赖一样，粗糙函数依赖的出现拓宽与丰

富了函数的研究领域。

本章在引入粗糙函数依赖基本定义的基础上系统地研究了粗糙函数依赖的性质和推理规则等，并分析了它与 Fuzzy 函数依赖、普通函数依赖的关系，本章的研究对于更好地利用 RRDB 有着重要的意义。可以看出，粗糙函数依赖的研究对应着粗糙关系数据库中决策规则的提取，其下近似定义的函数依赖相当于关系数据库的普通函数依赖，在知识发现中对应着确定性规则；而上近似定义的函数依赖对应着信息系统的部分依赖，在知识发现中对应着默认规则，虽然它并不象模糊函数依赖要求具有相同级别的相似度，但是其部分依赖程度越高，说明确定性规则占的比例越大。另外，在本章的基础上，可以进一步研究其规则提取等。

第 5 章　基于粗糙集与信息颗粒的聚类方法研究

淡泊明志，宁静致远
——诸葛亮《戒子篇》

5.1　引言

所谓聚类就是按照事物的某些属性，把事物聚集成类，使类间的相似性尽量小，类内的相似性尽量大。聚类是一个无监督的学习过程，分类是有监督的学习过程，两者的根本区别是：分类时需要事先知道分类所依据的属性值，而聚类则是要找到这个分类属性值[1]。进一步说，聚类的目的是发现样本点之间最本质的“抱团”性质，从信息粒度的角度看，聚类操作实际上是在一个统一的粒度下进行的计算，而分类操作是在不同的粒度之下进行的计算；总而言之，聚类是在一个均匀的、统一的粒度下来描述样本集，而分类是在非均匀粒度下来描述样本集上的先验知识[89]。

数据库聚类分析的目标是对象（或元组），每个对象由不同的属性构成，这些属性主要分为数字属性和字符属性，如果按照处理属性来划分的话，可以把数据库聚类的方法分为：（1）专门处理数字属性的方法；（2）专门处理字符属性的方法；（3）处理混合属性的方法——可以同时处理数字属性和字符属性。本章的方法主要是针对数据库混合属性处理的，考虑利用所有属性来进行自然的聚类分析。

本章首先对于当前的聚类方法进行了简单的小结；然后分析

了 Shoji Hirano 的基于粗糙集的聚类方法的不足，在此基础上提出了利用模糊隶属度、信息颗粒与粗糙集理论相结合进行数据库混合数据聚类的改进方法。具体地，讨论了信息颗粒和信息粒度的基本原理，提出了字符颗粒和数字颗粒的概念，研究了采用信息颗粒和粗糙集理论进行聚类的机理，并且给出了对纯字符数据和混合数据进行聚类的算法，每种算法又分自然聚类和按照要求的聚类数聚类。

5.2 聚类方法简述

聚类是把数据集合中相似的对象进行分组，把数据利用一些较少的类表示而忽略一些小的细节，目的是为了简化问题的处理，从数据建模的角度讲，聚类源于统计学与数字分析学；从机器学习的角度看，聚类对应于隐藏的模式，聚类的搜索是一种无监督学习；从实践的角度讲，聚类在数据挖掘应用中如科学的数据探索、信息检索、文本挖掘、空间数据库应用、Web 分析和医疗诊断等方面扮演着重要的角色[138]。聚类在统计学、模式识别和机器学习中是一个主要研究课题，由于聚类技术的分类不统一，而且方法繁多，将对聚类技术在数据挖掘中的应用提供一个大体的轮廓。

数据挖掘是一个与统计学、机器学习、数据库、逻辑程序设计和可视化等技术密切相关的一个研究学科，而聚类分析技术是知识发现中与分类、关联规则挖掘并列的基本技术。聚类技术与许多学科的关系很密切，经常用于统计学科中[138]，在模式识别中，聚类有很多应用，典型的应用是语音和特征识别[139]；在机器学习中，聚类算法被应用于图像分割和计算机视觉处理中；聚类技术也广泛应用于图像处理的数据压缩中，如矢量量化技术（Vector Quantization）等[140]。

对于聚类算法的分类，不同的文献分法不同，它们有时是相

互交迭的，这里在文献[141]和[1]的基础上给出一个简单的分法。

聚类方法分类：

（1）分层方法（Hierarchical Methods）。

- 聚集算法（Agglomerative Algorithms）。
- 分裂算法（Divisive Algorithms）。

基本的分层算法包括 Lance-Williams 公式法、概念聚类、SLINK、COBWEB、CURE 和 CHAMELEON 等。

（2）划分方法（Partitioning Methods）。

- 重置算法（Relocation Algorithms）。
- 概率聚类（Probabilistic Clustering）。包括 EM 框架、SNOB 算法、AUTOCLASS 和 MCLUST 等；
- K-medoids 方法。包括 PAM 算法、CLARA、CLARANS 算法及其扩展等。
- K-means 方法。
- 基于密度的方法（Density-Based Algorithms），包括：

① 基于密度的连接聚类（Density-Based Connectivity Clustering）。

② 密度函数聚类（Density Functions Clustering）。

（3）基于栅格的方法（Grid-Based Methods）。包括 BANG、STING 和 WaveCluster 等算法。

（4）基于并发控制的分类数据的聚类方法（Methods Based on Co-Occurrence of Categorical Data）：包括 ROCK、SNN 和 CACTUS 等算法。

（5）基于约束的聚类方法（Constraint-Based Clustering）。

（6）机器学习中的聚类算法。

- 梯度下降与人工神经网络（Gradient Descent and Artificial Neural Networks）。
- 进化方法（Evolutionary Methods）。

（7）可升级的聚类算法（Scalable Clustering Algorithms）。

（8）高维数据算法（Algorithms For High Dimensional Data）。

- 子空间聚类（Subspace Clustering）：包括 CLIQUE、MAFIA、ENCLUS、OPTIGRID、PROCLUS 和 ORCLUS 等算法；
- 投影技术（Projection Techniques）。
- 并发聚类技术（Co-Clustering Techniques）。

（9）模糊聚类方法。模糊聚类分析的开创性工作是 Ruspini 于 1969 年做出的，常用的模糊聚类方法有系统聚类法、传递闭包法以及与此等价的最大支撑树的 Prim 算法及 Kruskal 算法、动态直接聚类法、基于摄动的模糊聚类方法 FCMBP、模糊 C-均值法、模糊 ISODATA 算法和人工神经网络模糊聚类法[1]。

新的聚类分析技术还在不断出现，从聚类分析的发展来看，未来的聚类分析技术将向着能够处理大型的高维数据集和高噪声数据集，并且能够具有较高的稳定性方向发展。

5.3 基于粗糙集聚类方法的分析

Shoji Hirano 和 Shusaku Tsumoto 等在文献[142]中提出了利用粗糙集方法原理来进行聚类分析的思想，其思想可以归结为：

（1）对于每个对象分配一个初始的等价关系。

（2）利用阈值修改初始的等价关系，使用不同的阈值迭代这个过程，可以获得理想的聚类结果。

该方法是与传统方法完全不同的一种聚类方法，它避免了传统方法的许多缺陷，如聚类过程依赖于处理对象的出现顺序和要求中心点（聚类中心）等问题，但是该方法也有其不足之处，在对于数据库两个元组进行相似性度量时，它采用了下列公式：

$$S(x_i,\ x_j)=\frac{P_C}{P}\left(1-\frac{d_M(x_i,x_j)}{\max_{u,v}\in U^d M(x_u,x_v)}\right)+\frac{p_d}{p}\left(1-\frac{d_H(x_i,x_j)}{\max_{x_u,x_v}\in U_d H(x_u,x_v)}\right)$$

其中 S（x_i，x_j）为对象 x_i 与 x_j 的相似度，$d_M(x_i,\ x_j)$为测量数字属性的“Mahalanobis 距离”，$d_H(x_i,\ x_j)$为测量字符属性的“Hamming 距离”，这里，作者对于数字属性采用了“Mahalanobis 距离”，Mahalanobis 曾在 1948 年提出了两个总体间的“距离”的度量[143]：

$$\Delta^2=\sum_{j,k=1}^{p}a_{jk}(u_{1_j}-u_{2_j})(u_{1_k}-u_{2_k})$$

按照惯例 α 是协方差矩阵的逆。对应的统计量是

$$D^2=\sum_{j,k=1}^{p}a_{jk}(\overline{x}_{1_j}-\overline{x}_{2_j})(\overline{x}_{1_k}-\overline{x}_{2_k})$$

其中，α_{jk} 可以从合并协方差中计算得出。统计量 D^2 具有相当大的理论意义，但是在聚类分析中把 D^2 作为各点之间距离的度量是不合适的，事实上，对于一对点来讲协方差阵是退化的，距离不存在，在聚类分析中利用总的协方差矩阵显然是以未证实的假定为依据的[143]，因此对于数字属性的相似度量采用“Mahalanobis 距离”并不合适；即便可以使用 Mahalanobis 距离，其计算复杂度也相对要高，并且在求阈值的过程中需要多次排序，这无疑增加了计算量，也增加了算法的复杂度。

1965 年，美国自动控制论专家扎德（L.A.Zadeh）引入了模糊集合这一概念，其基本思想是把普通集合中的绝对隶属关系加以扩充，使元素对“集合”的隶属度由只能取 0 和 1 这两个值，推广到可以取单位区间[0，1]中的任意数值，从而实现定量地刻画模糊性事物。

隶属度是表现元素对集合的隶属关系不确定性大小的数量指

标，它表示了元素集合的隶属关系的客观可能性，是对模糊现象的一种客观描述。

基于上述考虑，本书用模糊数学的隶属度表示两个元素对于模糊关系的隶属程度来描述元组间数字属性的相似性。

5.4 聚类分析中的粒度与粗集原理研究

信息颗粒（Information Granule）与信息粒度（Information Granularity）是两个不同但却密切相关的概念。前面已经给出了关于信息颗粒、信息颗粒化和粒化计算的概念，按照 Zadeh 的观点，信息颗粒是通过不可分辨性（Indistinguish Ability）、相似性（Similarity）、近似性（Proximity）或功能性（或函数性，Functionality）等来划分的对象的集合。

从本质上讲，“颗粒”即基本元素，信息颗粒是在基本集（粗糙集概念）中具有相同属性值的对象集合，或着说，信息颗粒是通过不可分辨性、相似性和函数性等来划分的对象的集合，一个基本颗粒相当于粗糙集的一个等价类，等价类也称为等价颗粒，比如决策规则的前件、后件、规则本身等就是一种颗粒，再如基本颗粒 $a=1 \wedge b=1$ 可以定义为：$\|a=1 \wedge b=1\|_{IS}=\{x \in U: a(x)=1 \& b(x)=1\}$。

信息的颗粒化相当于把原始复杂的问题分解为多个易管理的子问题，即把大颗粒分解为小颗粒，颗粒化问题随处可见，它是许多学科共同的研究课题。

在本章中采用粗糙集与粒化计算原理来对数据进行聚类分析，粗糙集从理论上给出了数据库上的等价关系划分，而粒化计算则把信息颗粒作为一个有意义的原子，强调了颗粒的计算是信息处理的基础，而信息颗粒则作为数据挖掘的一个基本单位。

聚类（簇）可以说是信息颗粒化的同义词，无论采用何种聚类算法，目的是为了找到数据的结构和隐藏的类（簇）——即数

据集合中的信息颗粒。之所以要采用粒度原理辅助粗糙集来进行聚类分析，主要是粗糙集中拥有知识 R 的智能体（人、机器等）不能将等价类$[u]_R$ 中的对象与 u 分辨，也就是说粗糙集对于论域的理解只能达到等价关系的一个等价类这种颗粒状的程度，而对颗粒内的对象是无法分辨的。可以把粗糙集的等价颗粒进一步细化为小的等价颗粒，即本书提出的字符颗粒和数字颗粒。单独利用粗糙集的不可分辨关系进行聚类，对于样本的分析可能会不全面，而利用信息粒度原理分析每个样本的粒度情况则比较全面。事实上，粗糙集与粒度原理是相辅相成的，粗糙集的粒度结构是对论域的一个等价颗粒划分，而更细的划分则是把大的等价颗粒划分为小的等价颗粒，这一点可以从下面的分析中看到。

定义 5.1[89]　设 R 为等价关系，R_1，$R_2 \in R$，如果对于任意的 x，$y \in U$，均有 $x\ R_1\ y \Rightarrow x\ R_2\ y$，那么称 R_1 比 R_2 细，记为 $R_1 \leqslant R_2$。

对定义中的 R_1，R_2 来说，也可以说 R_2 比 R_1 粗糙，从等价颗粒的角度看，较细的等价关系 R_1 比较粗的等价关系 R_2 将产生更小的颗粒，即对于所有的 $x \in U$，有$[x]_{R_1} \subseteq [x]_{R_2}$，$R_2$ 的每个等价颗粒是 R_1 的等价颗粒的并集。对于两个等价关系 $R_1 \subset R_2$，则有，$\underline{apr}_{R_2}(X) \subseteq \underline{apr}_{R_1}(X) \subseteq X$，$X \subseteq \overline{apr}_{R_1}(X) \subseteq \overline{apr}_{R_2}(X)$，相应地有，$\alpha_{R_2}(X) \leqslant \alpha_{R_1}(X)$，其中 α 为集合的精确度量。推而广之，对于嵌套的等价关系 $R_1 \subseteq R_2 \subseteq \cdots \subseteq R_m$ 来说，有 $[x]_{R_1} \subseteq [x]_{R_2} \subseteq \cdots \subseteq [x]_{R_m}$ 成立，因而有：$\underline{apr}_{R_m}(X) \subseteq \cdots \subseteq \underline{apr}_{R_2}(X) \subseteq \underline{apr}_{R_1}(X) \subseteq X$，$X \subseteq \overline{apr}_{R_1}(X) \subseteq \overline{apr}_{R_2}(X) \subseteq \cdots \subseteq \overline{apr}_{R_m}(X)$，对于其精确度则有：$\alpha_{R_m}(X) \leqslant \cdots \leqslant \alpha_{R_2}(X) \leqslant \alpha_{R_1}(X)$。可见，较细的等价关系继承了较粗的等价关系的某些性质，嵌套的等价关系是一种偏序格结构。

聚类操作实质上是在样本点之间利用不可分辨性或相似性定义一种等价关系。在当前的阈值尺度下，类中的任意两个样本点

是没有区别的，一个等价关系就定义了样本点的一个划分，它把样本点划分成一些子集，一个子集就对应着一个类[89]。

5.5 基于粗糙集与信息粒度的聚类方法

5.5.1 基本概念

定义 5.2（**信息粒度**）[144] 设 $K=(U, R)$为一知识库，$P \in R$为一等价关系，也称为知识。P 的粒度记为 $GD(P)$，$GD(P)=card(P)/card(U^2)$其中 $card(P)$表示 $P \subseteq U \times U$ 的基数。当 P 为相等关系时，P 的粒度达到最小值$|U|/|U^2|=1/|U|$，当 P 为论域关系时，P 的粒度达到最大值$|U^2|/|U^2|=1$，因此一般地有：$0 \leqslant GD(P) \leqslant 1$。

粒度本身是物理学的概念，指“微粒大小的平均度量”，而信息粒度则是指对信息和知识细化的不同层次的度量，知识（信息）的粒度越小，分辨能力越强。文献[144]给出了关于粒度计算的如下命题，并给予了证明。

命题 5.1 设 P 为知识库 $K=(U, R)$中的知识，$U/P=\{X_1, X_2, \cdots, X_n\}$，则 $GD(P)=\sum_{i=1}^{n}|X_i|^2/|U|^2$。

定义 5.3（**字符颗粒**） 由不可分辨关系或相似关系等生成的字符数据颗粒（等价类或近似等价类）为字符颗粒。

定义 5.4（**数字颗粒**） 由不可分辨关系或相似关系等生成的数字数据颗粒（等价类或近似等价类）为数字颗粒。

定义字符颗粒和数字颗粒，是因为在给定的阈值下，每个对象可以生成自己的等价类或相似类结构，即每个对象有自己的颗粒结构，它们与等价颗粒是有差别的，字符颗粒和数字颗粒既可以是等价颗粒，也可以是近似等价颗粒，这比不可分辨关系的等价类要更进了一步。

定义 5.5（**隶属度**）[145] 对于论域 U，定义模式（U，U）上的

模糊等价关系 $\mathrm{EQUAL_U}$，则序偶（x_1，x_2）的隶属度 $\mu_{\mathrm{EQ_u}}$（x_1，x_2）表明了论域 U 中的对象 x_1 与论域 U 中的对象 x_2 对于模糊等价关系 $\mathrm{EQUAL_U}$ 的隶属程度，0 剟$\mu_{\mathrm{EQ_u}}$ 1。若 U 的成员为数字值，定义其隶属度 $\mu_{\mathrm{EQ_u}}(x_1, x_2)=\dfrac{1}{1+b_x|x_1-x_2|}$，$b_x(b_x>0)$为调节因子，是为了控制不同域间的隶属值，即为了避免某些属性的值过大而屏蔽其他取值较小的属性对数据相似性测量的影响。

定义 5.6　设 x_i，x_j 是两个具有 m 个字符属性的对象，x_i，x_j 之间的不相似度量可以定义为两个对象对应的字符属性值间总的不匹配量，不匹配数越小，两个对象越相似，文献[117]给出了这种度量的形式化描述：

$$d(x_i, x_j)=\sum_{k=1}^{m}\delta(x_i[A_k], x_j[A_k])$$

这里 $\delta(x_i[A_k], x_j[A_k])=\begin{cases}0 & x_i[A_k]=x_j[A_k]\\ 1 & x_i[A_k]\neq x_j[A_k]\end{cases}$，$A_k$ 表示属性。

下面利用隶属度 $\mu_{\mathrm{EQ_u}}$ 与定义 5.6 来定义数据库混合属性的相似性。

定义 5.7（相似度）　定义在单属性 A 下 R 的两个元组（或对象）x_i，x_j 的相似度为 $S_A(x_i, x_j)=\mu_{\mathrm{EQ_u}}(x_i[A], x_j[A])$，若 $A=\{A_1, A_2, \cdots, A_K\}$ 由 K 个数字属性构成，那么两个元组（对象）x_i，x_j 的相似度为：$S_{\mathrm{A}}(x_i, x_j)=\min\limits_{A_l\in A} S_{\mathrm{A}_l}$（$x_i$, x_j），若数据库既有数字属性，又有字符属性，即 $A'=\{A_1, A_2, \cdots, A_K, A_{K+1}, \cdots, A_m\}$ 则把两个元组的相似度定义为：$S_{A'}(x_i, x_j)=\min\limits_{A_l\in A} S_{A_l}(x_i, x_j)+\sum\limits_{f=k+1}^{m}\delta(x_i[A_f], x_j[A_f])$，其中 A_l 为数字属性，这里，$\delta(x_i[A_f], x_j[A_f])=\begin{cases}1 & x_i[A_f]=x_j[A_f]\\ 0 & x_i[A_f]\neq x_j[A_f]\end{cases}$，取与定义 5.6 中相反的值。

定义 5.8[142] 对象 x_i 的初始等价关系定义为：$R_{x_i}=\{\{P_i\},\{U-P_i\}\}$，其中 $P_i=\{x_j|S(x_i, x_j)\geqslant S_i\}$，$S(x_i, x_j)$为 x_i 与 x_j 的相似度即（x_i, x_j）对于模糊关系的隶属程度，S_i 为所取的阈值。

上述的定义 5.8 将一个论域分为了相似类及不相似类两部分。

5.5.2 基于粗糙集与信息颗粒的聚类方法

在分析上述基本概念的基础上，结合 Shoji Hirano 的基于粗糙集的聚类方法，提出基于粗糙集与信息颗粒的改进数据库混合数据的聚类方法如下：

（1）求数据库的字符颗粒集结构 U/R_C。即利用粗糙集的等价关系原理形成字符属性等价类及论域的字符颗粒结构。由于粗糙集理论擅长利用不可分辨关系处理字符或离散数据，因此首先对数据库的字符属性利用不可分辨关系进行字符属性等价类的划分，得到每个对象的字符颗粒结构（等价类划分）：R_{x_1}，R_{x_2}，…，R_{x_n}，取所有字符颗粒结构的交集便得到论域的字符颗粒集结构 $U/R_C=R_{x_1}\cap R_{x_2}\cap\cdots\cap R_{x_n}$，这一步实际上是按粗糙集的不可分辨关系原理得到论域的字符数据的自然划分。

（2）求数据库的数字颗粒集结构 U/R_n。即按照模糊隶属度定义的相似度上的不可分辨关系形成数字属性等价类及论域的数字颗粒结构：设 $U=\{x_1, x_2, \cdots, x_n\}$，对于数据库中的数字属性，按照模糊隶属度定义的相似度 $S_A(x_i, x_j)=\min\limits_{A_l\in A} S_{A_l}(x_i, x_j)$求出每个对象 x_i 与所有其他对象的相似度，指定相应的阈值，这里设阈值为 S_i，则某个对象的等价类或颗粒结构为$[x_i]=\{x'：S(x_i, x')\geqslant S_i\}$，由此形成等价关系 $R_{x_i}=\{[x_i], U-[x_i]\}$，这样由 n 个对象形成 n 个等价关系即一个等价关系簇$\{R_{x_1}, R_{x_2}, \cdots R_{x_n}\}$，根据粗糙集原理，取它们的交集便得到论域的数字颗粒划分 $U/R_n=R_{x_1}\cap R_{x_2}\cap\cdots\cap R_{x_n}$。

（3）初始等价关系 U/R 的形成。按照粗糙集原理，设属性集为 A，则有 $\mathrm{IND}(A)=\bigcap_{R\in A}\mathrm{IND}(R)$，因此初始的聚类结果为字符颗粒集等价关系 $U/R_C=R_{x_1}\cap R_{x_2}\cap\cdots\cap R_{x_n}$ 和数字属性颗粒集等价关系 $U/R_n=R_{x_1}\cap R_{x_2}\cap\cdots\cap R_{x_n}$ 的交集，即 $U/R=U/R_C\cap U/R_n$。

（4）最终等价关系即最终聚类结果 U/R' 的形成。初始等价关系形成后，需要进行聚类的优化，即根据需要进行类的调整（如按指定的类数进行聚类或进行自然聚类），目的为了取消小的聚类，使得聚类更加合理。若数据库为混合数据，则首先求出较小的类中所有对象与其余所有类的全部对象之间的距离，然后将较小的类中元素合并到与其距离最大（取相似度最大或平均距离最大）的类中，重复该过程，直到满足所要求的条件，这样便得到一种新的聚类结果。这里其调整用到本文的距离公式 $\min\limits_{A_f\in A}S_{A_f}(x_i,\ x_j)+\sum\limits_{f=k+1}^{m}\delta(x_i[A_f],x_j[A_f])$；若数据库为纯字符数据则可根据所定义的类间的可分辨程度，按照平均距离最大的原理和粗集原理对字符数据进行自然聚类，具体算法见 5.5.3 节。

上述步骤中（1）为按照粗糙集原理来形成字符属性的自然分类，步骤（2）为按照粗糙集原理形成数字属性的分类，它们是步骤（3）初始聚类结果形成的基础，步骤（3）利用粗糙集原理形成初始的聚类结果，步骤（4）为对初始等价关系的调整以形成最终的聚类结果。

5.5.3　算法描述

在文献[142]中测试数据库为布尔数据库（Balloon Database），而布尔数据库为纯字符数据库，因此在这里给出基于粗集原理的纯字符数据聚类的聚类算法及对于混合数据进行聚类的算法，每种聚类算法又分自然聚类和按照要求的聚类数进行聚类两种。

1．基于粗集与信息颗粒原理的纯字符数据的聚类

1）纯字符数据的自然聚类

聚类思想：首先按照粗集等价关系原理对数据库所有数据进行自然的分类，然后利用粒化原理（即信息颗粒原理），按顺序求每个类与其距离最大（平均距离最大）的类，把它们与其相似度最大的类进行合并，重复此过程直至无距离最大的类存在为止。

算法 5.1 数据库字符数据的自然聚类

（1）input S（m 行，n 列）；

（2）求 S_b=IND(A)={S_1，S_2，…，$S|S_b|$}；

（3）dMAX=0

（4）for i=1 to $card(S_b)$

（5）begin

（6）for j=i+1 to $card(S_b)$

（7）begin

（8）求类 S_i 与 S_j 之间的距离 DIS(S_i，S_j)，

（9）dMAX=max(dMAX，DIS(S_i，S_j))；

（10）end

（11）end

求类间的最大距离 dMAX，DIS（S_i，S_j）为两个类 S_i 与 S_j 之间的距离，即 S_i 中所有元素与类 S_j 距离的平均值，S_i 中每个元素 d_f 与 S_j 的距离为 d_f 与 S_j 中每个元素 d_g 距离和 sum 的平均值，即 DIS(u，S_j)=sum/$card(S_j)$，d_f，d_g 的距离为 $S_A(d_f, d_g)=\sum_{k=1}^{n}\delta(d_f[A_k], d_g[A_k])$

下面求与每个类距离为 dMAX 的第一个类，并与之合并

（12）FOR i=1 to $card(S_b)$

（13）Begin

（14）For j=i+1 to $card(S_b)$

（15）Begin

（16）计算 S_i 与 S_j 的距离 DIS（S_i，S_j）

（17）if DIS（S_i，S_j）=dMAX

（18）$S_i=S_i \cup S_j$，

（19）$card(S_b)= card(S_b)-1$；

（20）Break；

（21）End

（22）break

（23）End

（24）输出聚类结果 S_z 及聚类数。

2）算法举例

以 BALLOON 数据库为例，如表 5-1（URL：http://www.ics.uci.edu/ pub/machine-learning-databases ）

（1）按照粗集不可分辨关系（等价关系）可得：IND(color，size，act，age，infl)={{V_1，V_5}，{V_2，V_6}，{V_3，V_7}，{V_4，V_8}，{V_9}，{V_{10}}，{V_{11}}，{V_{12}}，{V_{13}}，{V_{14}}，{V_{15}}，{V_{16}}，{V_{17}}，{V_{18}}，{V_{19}}，{V_{20}}}，共计 16 类。

（2）计算类间的距离，可得最大为 4，按照距离为 4 进行类的合并，可以得到与{V_1，V_5}距离为 4 的第一个类为{V_2，V_6}，则合并得：

{{V_1，V_5，V_2，V_6}{V_3，V_7}，{V_4，V_8}，{V_9}，{V_{10}}，{V_{11}}，{V_{12}}，{V_{13}}，{V_{14}}，{V_{15}}，{V_{16}}，{V_{17}}，{V_{18}}，{V_{19}}，{V_{20}}}。

（3）判定是否有与{V_1，V_5，V_2，V_6}距离为 4 的类，若无，顺次计算{V_3，V_7}，与其距离最大的第一个类为{V_4，V_8}，合并可得到：

{{V_1，V_5，V_2，V_6}，{V_3，V_7，V_4，V_8}，{V_9}，{V_{10}}，{V_{11}}，{V_{12}}，{V_{13}}，{V_{14}}，{V_{15}}，{V_{16}}，{V_{17}}，{V_{18}}，{V_{19}}，{V_{20}}}。

表 5-1　BALLOONS 数据库

Obj	color	size	act	age	infl
V_1	YELLOW	SMALL	STRETCH	ADULT	T
V_2	YELLOW	SMALL	STRETCH	CHILD	T
V_3	YELLOW	SMALL	DIP	ADULT	T
V_4	YELLOW	SMALL	DIP	CHILD	T
V_5	YELLOW	SMALL	STRETCH	ADULT	T
V_6	YELLOW	SMALL	STRETCH	CHILD	T
V_7	YELLOW	SMALL	DIP	ADULT	T
V_8	YELLOW	SMALL	DIP	CHILD	T
V_9	YELLOW	LARGE	STRETCH	ADULT	F
V_{10}	YELLOW	LARGE	STRETCH	CHILD	F
V_{11}	YELLOW	LARGE	DIP	ADULT	F
V_{12}	YELLOW	LARGE	DIP	CHILD	F
V_{13}	PURPLE	SMALL	STRETCH	ADULT	F
V_{14}	PURPLE	SMALL	STRETCH	CHILD	F
V_{15}	PURPLE	SMALL	DIP	ADULT	F
V_{16}	PURPLE	SMALL	DIP	CHILD	F
V_{17}	PURPLE	LARGE	STRETCH	ADULT	F
V_{18}	PURPLE	LARGE	STRETCH	CHILD	F
V_{19}	PURPLE	LARGE	DIP	ADULT	F
V_{20}	PURPLE	LARGE	DIP	CHILD	F

重复上述过程，最后得到：{{V_1，V_5，V_2，V_6}，{V_3，V_7，V_4，V_8}，{V_9，V_{10}}，{V_{11}，V_{12}}，{V_{13}，V_{14}}，{V_{15}，V_{16}}，{V_{17}，V_{18}}，{V_{19}，V_{20}}}，该聚类中已无类间距离为 4 的类，因此为最后的聚类结果，聚类数为 8。

其粒度为：2×(16/400)+6×(4/400)=0.14。

把类内元素的不可分辨程度定义为类内所有元组的交集的元素个数与属性个数之比，即$\gamma(S)=card(\bigcap_{i=1}^{|S|} x_i)/card(A)$，其中 $card(A)$

为属性数，$\bigcap_{i=1}^{|S|} x_i$ 为所有类内元素 x_i 对应属性值的交集。在第一步聚类中，对象间实际的不可分辨程度为 100%，在后边的调整中，实际上是调整类内元素的不可分辨程度，如上例中，按照类间最大距离为 4，这时类内元素的的最大不可分辨程度为 4/5=80%。

2．按照要求的聚类数对字符数据进行聚类

聚类思想：首先按照粗集等价关系原理对数据库所有数据进行自然分类，然后选出第一个最小的类，求该类与其距离最大（取平均距离最大）的类，把它与其相似度最大的类合并，判定是否满足所要求的聚类数，若不满足，则再选出较小的类，重复上述过程，若满足所要求的聚类数结束。

算法 5.2　按要求的聚类数进行聚类：

（1）input S (m 行，n 列)，K//K 为聚类数

（2）S_b=IND(A)={S_1，S_2，…，$S_{|S_b|}$}，设其类数为 $card(S_b)$

（3）if $card(S_b)<K$ then 按等价关系分解为指定的类数 K

（4）if $card(S_b)=K$ then goto （22）

（5）if $card(S_b)>K$ then

（6）for i=1 to card(S_b)

（7）begin

（8）求 $card(S_i)$，

（9）找出元素数最少的第一个类 S_{small}；

（10）end

（11）SM=S_{small}；

（12）SS=0

（13）dis(S_i，S_j)= dis(S_j，S_i)；

（14）For i=1 to $card(S_b - S_{small})$

（15）begin

（16）求与 S_{small} 距离最大的第一个类 S_p

（17）end

（18）标注平均距离最大的类 $S_{max}=S_P$；

（19）$S_{small}=S_{small}\cup S_p$；

（20）if 总聚类数等于 K goto (22)

（21）if 总聚类数大小 K 转（6）

（22）输出最后的聚类结果 S_e 及聚类数 K。

举例： 要求把布尔数据库分为 7 类：

（1）按照不可分辨关系可得：

IND（color，size，act，age，infl）={{V_1，V_5}，{V_2，V_6}，{V_3，V_7}，{V_4，V_8}，{V_9}，{V_{10}}，{V_{11}}，{V_{12}}，{V_{13}}，{V_{14}}，{V_{15}}，{V_{16}}，{V_{17}}，{V_{18}}，{V_{19}}，{V_{20}}}。

（2）选出第一个小类{V_9}，计算{V_9}与其他类的距离，可得最大为 4 的第一个类为{V_{10}}按照距离为 4 进行类的合并，可以得到{V_9，V_{10}}。

（3）判断类数，为 15 类，不满足要求的 7 类，则再将{V_{11}}选出，与其距离最大的为 V_{12}，合并得{V_{11}，V_{12}}，判断聚类数为 14，不满足聚类为 7 的要求，此时分类为：

{{V_1，V_5}，{V_2，V_6}，{V_3，V_7}，{V_4，V_8}，{V_9，V_{10}}，{V_{11}，V_{12}}，{V_{13}}，{V_{14}}，{V_{15}}，{V_{16}}，{V_{17}}，{V_{18}}，{V_{19}}，{V_{20}}}。

依次进行，最后聚类结果为：{V_1，V_5，V_2，V_6}，{V_3，V_7，V_4，V_8}，{V_9，V_{10}，V_{11}，V_{12}}，{V_{13}，V_{14}}，{V_{15}，V_{16}}，{V_{17}，V_{18}}，{V_{19}，V_{20}}。

其划分粒度为：3×16/400+4×4/400=0.16

3．基于粗集与粒化原理的数据库混合数据的聚类

即数据库中既有数字属性数据，又有字符属性数据，其聚类算法设计仍然按照自然聚类与按要求的聚类数聚类两种思路进行。

1）混合数据的自然聚类

聚类思想：首先对数据库的字符数据按粗集等价关系原理进行自然聚类形成字符颗粒结构，然后按照粗集与粒化原理，按照模糊隶属度定义的相似度对数字数据进行聚类形成数字颗粒结构，取字符颗粒结构与数字颗粒结构的交集形成初始的等价关系即原始的聚类结构；接着进行类的优化处理，选出较小的类集，求这些类中每个元素与其最接近的大类并将其依次合并入内，直到无规定的小类为止形成最终的聚类结果。

算法 5.3　数据库混合数据的自然聚类：

（1）input $S(m$ 行，n 列)，设 S 有 h 个数字属性，$n–h$ 个字符属性，设 $U=\{x_1, x_2, \cdots, x_n\}$。

（2）设有 $n–h$ 个字符属性，$R/S_z=\mathrm{IND}(A)=\mathrm{IND}(A_{h+1}, A_{h+2}, \cdots, A_n)$。

（3）对于数据库中的数字属性，按照模糊隶属度定义的相似度 $S_A(x_i, x_j)=\min\limits_{A_l \in A} S_{Al}(x_i, x_j)$求出每个对象 x_i 与所有其他对象的相似度。

（4）$S_A(x_i, x_j)=S_A(x_j, x_i)$

（5）for i=1 to m

（6）for $j=i$+1 to m

（7）begin

（8）给 b_x 赋值；for k=1 to h

（9）$S_{A_k}(x_i, x_j)=\dfrac{1}{1+b_x|x_i[A_k]-x_j[A_k]|}$

（10）SS= $S_{A_1}(x_i, x_j)$

（11）for k=2 to h

（12）begin

（13）$S_A(x_i, x_j)=\min\limits_{A_k \in A}\{SS, S_{A_k}(x_i, x_j)\}$

（14）SS= $S_A(x_i, x_j)$

（15）End

（16）end

（17）设阈值为 T，则$[x_i]=\{x':S(x_i,\ x')\geqslant T\}$

（18）$R_{x_i}=\{[x_i],\ U-[x_i]\}$

（19）for i=1 to m

（20）$R/x_i=\{[x_i],\ U-[x_i]\}$

（21）$U/S_n=R_{x_1}\cap R_{x_2}\cap\cdots\cap R_{x_n}$

（22）$U/R=U/S_n\cap R/S_z$

（23）下面为优化处理，即对原始聚类的优化

（24）$S_{small}=\{\ \ \}$

（25）For i=1 to $card(U/R)$

（26）If $card(S_i)\leqslant\varepsilon$ $S_{small}=S_{small}\cup S_i$；元素个数≤ε的类为较小的类

（27）SS1=0

（28）DIS=0

（29）Let DIS(S_i，S_j)= DIS(S_j，S_i)

（30）For i=1 to $card(S_{small})$

（31）begin

（32）For f=1 *to* $card(S_i)$

（33）begin

（34）FOR j=1 to $card\{U/R-S_{small}\}$

（35）Begin

（36）For g=1 to $card(T_j)$

（37）begin

（38）$S_A(d_f,d_g)=\min\limits_{A_l\in A}R_{A_l}(d_f,d_g)+\sum\limits_{k=h+1}^{n}\delta(x_i[A_k],x_j[A_k])$

（39）DIS(d_f，T_j)=max(DIS，$S_A(d_f,\ d_g)$)

（40）DIS=DIS(d_f，T_j)

（41）end

（42）End

（43）d_f 为 S_i 的第 f 个元素，d_g 为 T_j 的第 g 个元素，通过上面循环的内双重循环求出与 d_f 相似度最大的类为 T_j，下面把 d_f 合并到 T_j 中

（44）$T_j = T_j \cup \{d_f\}$

（45）End

（46）End

（47）输出最后的聚类结果 S_e 及聚类数 K。

2）算法举例

下面以文献[142]中的例子（表 5-2）来说明上面的方法。

表 5-2 中论域为 $U=\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$，属性集 $A=\{A_{tt1}, A_{tt2}, A_{tt3}, A_{tt4}\}$，设等价关系为 R，对其进行聚类分析如下：

首先对字符属性按不可分辨关系分析每个对象与其余对象的关系，现字符属性集 $A_1=\{A_{tt3}, A_{tt4}\} \subseteq A$，则得到每个对象的颗粒结构：

$R_{x_1} = R_{x_2} = R_{x_3} = R_{x_4} = \{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7, x_8, x_9\}\}$（按等价关系分为等价的与不等价的两类），$R_{x_5} = \{\{x_5\}, \{x_1, x_2, x_3, x_4, x_6, x_7, x_8, x_9\}\}$，$R_{x_6} = R_{x_7} = R_{x_8} = R_{x_9} = \{\{x_6, x_7, x_8, x_9\}, \{x_1, x_2, x_3, x_4, x_5\}\}$。

则字符颗粒结构 $U/R_C = R_{x_1} \cap R_{x_2} \cap R_{x_3} \cap R_{x_4} \cap R_{x_5} \cap R_{x_6} \cap R_{x_7} \cap R_{x_8} \cap R_{x_9} = \{\{x_1, x_2, x_3, x_4\}, \{x_5\}, \{x_6, x_7, x_8, x_9\}\}$。

表 5-2　混合数据举例

Obj	A_{tt1}	A_{tt2}	A_{tt3}	A_{tt4}
x_1	0	0	Round	Small
x_2	0.1	0	Round	Small

续表

Obj	A_{tt1}	A_{tt2}	A_{tt3}	A_{tt4}
x_3	0	0.1	Round	Small
x_4	0.1	0.1	Round	Small
x_5	0.15	0.15	Square	Small
x_6	0.3	0.3	Square	Large
x_7	0.4	0.3	Square	Large
x_8	0.3	0.4	Square	Large
x_9	0.4	0.4	Square	Large

其次计算数字颗粒集，数字属性集 $A_2=\{A_{tt1}, A_{tt2}\}\subseteq A$，按照公式 $S_A(x_i, x_j)=\min\limits_{A_l\in A} S_{A_l}(x_i, x_j)$，设 $b_x=1$ 计算得到每个对象的颗粒结构：对于 x_1 来说，它与其余 8 个对象的相似度分别为：$S_{A_2}(x_1, x_2)=\min\{1/1.1, 1\}=0.91$（取两位小数），$S_{A_2}(x_1, x_3)=S_{A_2}(x_1, x_4)=0.91$，$S_{A_2}(x_1, x_5)=0.87$，$S_{A_2}(x_1, x_6)=0.77$，$S_{A_2}(x_1, x_7)=S_{A_2}(x_1, x_8)=S_{A_2}(x_1, x_9)=0.71$，如果取相似度阈值为 $S_i\geqslant 0.91$（注：根据需要取不同的阈值将得到不同的聚类结果），则得到 x_1 的颗粒结构 $R_{x_1}=\{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7, x_8, x_9\}\}$，类似地，可以得到其余对象的颗粒结构：$R_{x_2}=R_{x_3}=\{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7, x_8, x_9\}\}$，$R_{x_4}=\{\{x_1, x_2, x_3, x_4, x_5\}, \{x_6, x_7, x_8, x_9\}\}$，$R_{x_5}=\{\{x_4, x_5\}, \{x_1, x_2, x_3, x_6, x_7, x_8, x_9\}\}$，$R_{x_6}=R_{x_7}=R_{x_8}=R_{x_9}=\{\{x_6, x_7, x_8, x_9\}, \{x_1, x_2, x_3, x_4, x_5\}\}$。

则数字颗粒集 $U/R_n=R_{x_1}\cap R_{x_2}\cap R_{x_3}\cap R_{x_4}\cap R_{x_5}\cap R_{x_6}\cap R_{x_7}\cap R_{x_8}\cap R_{x_9}=\{\{x_1, x_2, x_3\}, \{x_4\}, \{x_5\}, \{x_6, x_7, x_8, x_9\}\}$。

因此初始的聚类结果为 $U/R=U/R_C\cap U/R_n=\{\{x_1, x_2, x_3\},$

$\{x_4\}$，$\{x_5\}$，$\{x_6, x_7, x_8, x_9\}\}$。

考虑到$\{x_4\}$，$\{x_5\}$单独作为一类比较小，可以把它们合并到其余类中，按照上面的距离公式计算 x_5 与其余 7 个对象的距离，得到：$S_A(x_5, x_1) = S_A(x_5, x_2) = S_A(x_5, x_3)$ =0.87+0.1=1.87，$S_A(x_5, x_6)$ =1.87，$S_A(x_5, x_7) = S_A(x_5, x_8) = S_A(x_5, x_9)$ =1.8。因此按照最大距离法(平均距离最大)，把$\{x_5\}$合并到$\{x_1, x_2, x_3\}$中，x_4 同样也可以并入$\{x_1, x_2, x_3\}$中，最后的聚类结果为：

$U/R=\{\{x_1, x_2, x_3, x_4, x_5, \}, \{x_6, x_7, x_8, x_9\}\}$。

需要说明的是，进行聚类时所取的阈值不同，所得到的初始聚类结果便不同，但实验表明它不影响经过聚类优化后的最终结果，这是由于本文所采用的数字距离公式的鲁棒性和粗糙集的不可分辨能力所致。关于阈值的取法，可以按照经验来取，也可以按照相应的阈值函数来取，或按照多级阈值来处理，这里就不做详细讨论了，当阈值相对比较大时，则聚类比较粗糙，当阈值比较小时，则聚类比较细致。

4．按照指定的聚类数对混合数据进行聚类

聚类思想：首先对数据库的字符数据按粗集等价关系原理进行自然聚类形成字符颗粒结构，然后按照粗集与粒化原理，按照模糊隶属度定义的相似度对数字数据进行聚类形成数字颗粒结构，取字符颗粒结构与数字颗粒结构的交集形成初始的等价关系即原始的聚类结构；接着进行类的优化处理，选出较小的类集，求这些类中每个元素与其最接近的大类并将其依次合并入内，每合并完一个类中所有元素便判断是否满足所要求的聚类数，直到满足规定的聚类数形成最终的聚类结果。

算法 5.4

（1）input $S(m$ 行，n 列)，K，设 $U=\{x_1, x_2, \cdots, x_n\}$

（2）$A=(A_{h+1}, A_{h+2}, \cdots, A_n)$，$S_b$=IND($A$)，设其类数为 $card(S_b)$

（3）对于数据库中的数字属性，按照模糊隶属度定义的相似度 $S_A(x_i，x_j)=\min\limits_{A_l\in A} S_{A_l}(x_i，x_j)$求出每个对象 x_i 与所有其他对象的相似度

（4）$S_A(x_i，x_j)=S_A(x_j，x_i)$

（5）for i=1 to m

（6）for $j=i$+1 to m

（7）begin

（8）begin

（9）给 b_x 赋值；for k=1 to h

（10）$S_{A_k}(x_i，x_j)=\dfrac{1}{1+b_x|x_i[A_k]-x_j[A_k]|}$

（11）end

（12）SS=$S_{A_1}(x_i，x_j)$

（13）for k=1 to h

（14）$S_A（x_i，x_j）=\min\limits_{A_k\in A}\{\mathrm{SS}，S_{Ak+1}(x_i，x_j)\}$

（15）SS=$S_A（x_i，x_j）$

（16）end

（17）设阈值为 T，$[x_i]=\{x':S(x_i，x')\geqslant T\}$，

（18）for i=1 to m

（19）$R/x_i=\{[x_i]，U-[x_i]\}$

（20）$U/S_n=R_{x_1}\cap R_{x_2}\cap\cdots\cap R_{x_n}$

（21）$U/R=U/S_n\cap R/S_z$

（22）if $card(S_b)<K$，则按等价关系分解为指定的类数 K

（23）if $card(S_b)=K$，则已完成指定的要求，转（46）

（24）if $card(S_b)>K$ then

（25）SS=0

（26）for i=1 to $card(U/R)$

（27）begin

（28）$card(S_{small})$=min(SS，$card(S_i)$)，其中 $S_i \subseteq U/R$

（29）SS= $card(S_{small})$

（30）end

（31）SMIN=S_{small}

（32）SS=0

（33）dis(S_i，S_j)= dis(S_j，S_i)

（34）for f=1 to $card(S_{small})$

（35）begin

（36）for i=1 to $card(U/R–S_{small})$

（37）begin

（38）求 U/R 中 S_{small} 元素与其他类的距离 dis(x_f，S_i)

（39）标注与 x_f 距离最大的现存类 S_{max}={S_p}

（40）S_p=S_p∪{x_f}

（41）end

（42）end

（43）其中 S_i，为初始等价关系 U/R 的类，dis(x_f，S_i)按算法 5.3 的定义

（44）if 总聚类数=K goto(46)

（45）else 总聚类数>K goto(25)

（46）输出最后的聚类结果 S_e 及聚类数 K

5.5.4　实验比较与分析

下面以 BALLOON 数据库和文献[142]的例子作为测试数据来对本文的聚类方法与文献[142]的方法做以比较。

现要求的聚类数为 7，文献[142]与本文算法对于 BALLOON 数据库进行聚类的结果为：

比较结果见表 5-3，显然，利用本文的方法所得的聚类结果比文献[142]更加合理一些，文献[142]将 V_{17}，V_{18}，V_{19}，V_{20} 各分

为四类是不合理的，因为它们之间很明显相似度比较大，本文聚类结果{V_1，V_5，V_2，V_6}，{V_3，V_7，V_4，V_8}，{V_9，V_{10}，V_{11}，V_{12}}，{V_{13}，V_{14}}，{V_{15}，V_{16}}，{V_{17}，V_{18}}，{V_{19}，V_{20}}则比较合理，符合类内相似度尽量大，类间相似度尽量小的聚类的基本原则：比如对于 V_{17}，V_{18}，其不可分辨程度为 80%，而把它们单独分成两类则不合理，其余{V_{19}，V_{20}}亦然。

表 5-3　结果比较

Obj	color	Size	act	Age	infl	文[113]	本文结果
V_1	YELLOW	SMALL	STRETCH	ADULT	T	1	1
V_2	YELLOW	SMALL	STRETCH	CHILD	T	1	1
V_3	YELLOW	SMALL	DIP	ADULT	T	1	2
V_4	YELLOW	SMALL	DIP	CHILD	T	1	2
V_5	YELLOW	SMALL	STRETCH	ADULT	T	1	1
V_6	YELLOW	SMALL	STRETCH	CHILD	T	1	1
V_7	YELLOW	SMALL	DIP	ADULT	T	1	2
V_8	YELLOW	SMALL	DIP	CHILD	T	1	2
V_9	YELLOW	LARGE	STRETCH	ADULT	F	2	3
V_{10}	YELLOW	LARGE	STRETCH	CHILD	F	2	3
V_{11}	YELLOW	LARGE	DIP	ADULT	F	2	3
V_{12}	YELLOW	LARGE	DIP	CHILD	F	2	3
V_{13}	PURPLE	SMALL	STRETCH	ADULT	F	3	4
V_{14}	PURPLE	SMALL	STRETCH	CHILD	F	3	4
V_{15}	PURPLE	SMALL	DIP	ADULT	F	3	5
V_{16}	PURPLE	SMALL	DIP	CHILD	F	3	5
V_{17}	PURPLE	LARGE	STRETCH	ADULT	F	4	6
V_{18}	PURPLE	LARGE	STRETCH	CHILD	F	5	6
V_{19}	PURPLE	LARGE	DIP	ADULT	F	6	7
V_{20}	PURPLE	LARGE	DIP	CHILD	F	7	7

算法 5.3 是对混合数据进行自然聚类，该算法是本章的主要

算法，利用该算法得到与文献[142]相同的结果，但是本文的算法复杂度要比其低。综合比较本文的方法与文献[142]的方法，可以得出下列结论。

（1）本章的方法简便，易于操作，算法复杂度低。方法充分利用了粗糙集原理的不可分辨关系处理字符属性，利用模糊关系下的隶属度定义数字属性的相似性来处理数字属性并按不可分辨关系对它分类，方法简便，可操作性强；从算法复杂度上看，文献[142]的算法时间复杂度为 $O(n(n\log n+2n+1))$，以算法 5.3 为例，本方法的时间复杂度为 $O(n+\frac{1}{2}n(n-1)+\frac{1}{2}n(n-1)+n^2+n)=O(n(2n+1))$，本文利用阈值对于数字属性进行分类后，在调整阶段不再与阈值发生关系，并且不需要进行排序过程，因此本文的算法要相对地节省时间，从比较可以看出本文的方法简明、实用并且算法复杂度低。

（2）本章方法分析全面，避免了一些传统聚类方法的一些缺陷，比如要求聚类中心，聚类依赖于对象顺序等。本方法中每个对象的地位是平等的，不需要聚类中心，每个对象均得到了兼顾，而且聚类的过程不依赖于对象排列的顺序，这点与文献[142]相同，但是由于采用了模糊等价关系与粗糙集等价关系原理相结合，因此本方法鲁棒性较好，算法抗噪声能力较强。

（3）利用本章方法可以方便地对聚类结果进行粒度分析，以考察聚类效果。粒度值越大，则聚类数目较少，聚类较粗糙，粒度越小，则聚类数目较多，聚类较细致。本文算法 5.3 例子中，根据前面的命题，最后的聚类结果粒度为 25/81+16/81=0.51，字符颗粒粒度为 16/81+1/81+16/81=0.41，数字颗粒粒度为 9/81+1/81+1/81+16/81=0.33，具体应用中，可根据对类数的要求调整粒度及颗粒结构。

本章在分析聚类原理的基础上提出了字符颗粒和数字颗粒的概念，并研究了把模糊方法与粗糙集、粒度原理结合起来进行聚类的方法，在数据库混合属性的聚类上做了一些有益的探索，文

中提出的方法具有普适性，进一步可以研究本方法与相关的处理混合属性的聚类方法的关系及相应的聚类函数依赖等；另外在相关的文献中也有先利用粗集进行数据约简，再进行数据聚类的，这是另外一种思路，这里不再加以讨论。

第 6 章　信息系统函数依赖的信息颗粒原理与计算

学而不思则罔，思而不学则殆
——孔子《论语》

6.1　引言

在一般的数据库知识发现系统中，需要考虑四个问题：依赖性分析（Dependency Analysis）、类的标识（Class Identification）、类的描述（Class Description）和偏差检测（Deviation Detection）[147]。可以说数据挖掘的任务之一就是对数据库的属性变量建立依赖性模型：即描述属性变量间重要的依赖关系，数据依赖性表示了可发现知识里重要的一类，而函数依赖是数据库数据依赖中最基本的依赖关系。

函数依赖性的研究有很多应用，它们常用于数据库的规范化及设计、查询优化、数据约简和规则提取中。在数据库知识发现系统中，依赖性的分析结果可以直接引起终端用户的兴趣，如揭示未知的属性函数依赖性，一般地，强依赖性反映了数据库固有的域结构。自动发现检测依赖性可以为模式抽取算法发现知识提供一个有益的方法。

学者 T.Y.Lin 多年来发表了系列论文研究与信息颗粒和颗粒计算有关的关系数据库面向机器的数据挖掘建模理论问题，提出了利用信息颗粒的位表示来进行数据挖掘的思想，他主要的工作是把该思想用于关系数据库的建模及各种关联规则的发现，其研

究方法为数据挖掘提供了一种新的思路。

本章在研究面向机器的数据挖掘模型的基础上提出一种利用信息颗粒位方式（或表示）进行信息系统函数依赖判定的方法。探讨了 T.Y.Lin 提出的信息颗粒位方式（或表示）及面向机器的数据挖掘模型，研究了关于位表示的性质，然后利用信息颗粒的位表示方法研究了信息系统的函数依赖、恒等依赖和部分依赖的信息颗粒原理，得出了它们相关的性质。该方法可以快速判定和度量信息系统的函数依赖关系，对于信息颗粒采用其位表示，使得数据格式更接近机器的内部表示，且能够直接挖掘信息系统所蕴含的各种模式。

6.2 面向机器的数据挖掘模型

6.2.1 模型语义

信息系统实际上是数据库关系概念的泛化，不同的是信息系统中不同的对象可以有相同的属性值（即两个元组完全相同），而数据库关系则不允许，数据库的所有关系（实例）受数据库范式的约束，而信息系统则不然。当信息系统中没有任意两个元组完全相同时，信息系统等同于数据库的关系，因此从某种程度上说，信息系统的很多原理也适合于数据库，它们是可以相互转化的。关系数据库理论更重视的是属性值本身，而信息系统则兼顾对象及其属性值。一般来说数据库系统按照块状数据的语义来处理数据（组织和排序等），它是一种面向人工的处理方式，它处理的主要对象是属性值，即属性值是人工方式的有意义的基本概念。

数据挖掘是从机器存储的数据中寻找隐藏的语义，即模式，属性值对机器来说只是位与字节而已，因此属性值对于机器处理并不方便。面向机器的处理是以基本颗粒为主体的，而基本颗粒

是论域（对象的集合）的一个子集，因此使用颗粒作为属性值的表示，亦即把对象子集（颗粒）作为主要处理对象，这就是面向机器的数据挖掘模型，它与人工方式（单属性值为主体）是不同的[148]。换句话说，在关系模型中，从机器的语义讲，属性值是一个有意义的对象集合的标识，该集合被称为论域的颗粒，颗粒本身也可以被认为是集合（颗粒）的标识，该标识被称为颗粒的规范名（Canonical Name），关系模型使用这些规范名来作为属性值称为面向机器的数据挖掘模型。

6.2.2　信息颗粒的位表示

由于属性值的机器语义是一些对象的集合，因而可以用颗粒的位模式来取代属性值。按照粗集原理，论域被划分为互不相交的，并集为论域的非空子集，每个非空子集被称为等价类，等价类的集合称为商集，记为 U/R，每个等价类充当两个角色：商集的一个元素和论域的一个子集。作为商集的一个元素，称之为规范名，它是第二个角色子集的标识，因此可以说商集是由规范名构成的，每个规范名表示的是一个等价类[149]。

在一个信息系统中，设 U 为论域，C 为基本概念的集合（属性值），m 为 U 到 C 的映射，m: $U\rightarrow C$，则（x，$m(x)$）称为信息表，它兼顾对象及其对应的属性值，而关系数据库则只关系属性值本身，即 $m(x)$。

下面通过举例来研究信息系统的基于信息颗粒及位方式的性质。以信息表 6-1[148]为例。

表 6-1 为一个与供应商相关的信息表，表 6-2 为表 6-1 的部分属性值的颗粒中心和位表示。

定义 6.1　设 BIT 为映射函数，BIT：$M\rightarrow \mathrm{bin}_1\mathrm{bin}_2\cdots\mathrm{bin}_i\cdots \mathrm{bin}_{|U|}$，其中 $M=\{v_1，\cdots，v_i，\cdots，v_j，\cdots\}$为属性值的等价颗粒中心，其映射值为：当 $v_i\in M$ 时，$\mathrm{bin}_i=1$，当 $v_i\notin M$ 时，$\mathrm{bin}_i=0$。

表 6-1　供应商表

U	$S\#$	Sname	Status	City
v_1	S_1	Smith	Twenty	C_1
v_2	S_2	Jones	Ten	C_2
v_3	S_3	Blake	Ten	C_2
v_4	S_4	Clark	Twenty	C_1
v_5	S_5	Adams	Thirty	C_3

表 6-2　信息颗粒的位表示

属性值	颗粒中心	位表示
Twenty	v_1，v_4	10010
Ten	v_2，v_3	01100
Thirty	v_5	00001
C_1	v_1，v_4	10010
C_2	v_2，v_3	01100
C_3	v_5	00001

根据信息颗粒原理分析由属性 status 形成的等价颗粒中心与位表示形式。对于 status，它由三个属性值构成：Twenty，Ten，Thirty，在该表中，属性值为 Twenty 的对象集合用 NAME（Twenty）表示，则 NAME（Twenty）=$\{v_1, v_4\}$，它的位表示为 10010，记为 BIT（Twenty）=10010，而 NAME（Ten）=$\{v_2, v_3\}$，其位表示为 BIT（Ten）=01100，NAME（Thirty）=$\{v_5\}$，其位表示为 BIT（Thirty）=00001，显然，Twenty，Ten，Thirty 构成论域 U 的一个划分{10010，01100，00001}，其中 NAME（Twenty）为规范名表示，从上面的分析得出信息系统单列的面向机器的关系模型是[148]：

（1）论域的划分：比如{Twenty，Ten，Thirty}为 Status 列的划分。

（2）每个等价类是一个给定的规范名，比如 Twenty。

（3）每个对象属于一个规范名，比如 v_1 属于 Twenty 等价类。

类似地，一个多列的面向机器的数据挖掘模型为：

（1）论域的划分簇。

（2）每个等价类是一个给定的规范名。

（3）对于每个划分来说，存在一个映射分配每个对象到相应的规范名中。

6.3　位表示的性质研究

首先对本章所涉及的基本概念做简单的介绍与解释。

定义 6.2[148]　设二元关系 B^i 是一个子集，$B^i \subseteq V \times U$，它定义 $p \in V$（V 为对象空间，U 称为数据空间）的二元（基本）邻域为 $B^i{}_P=\{u|u\ B^i\ p\}$，映射 β^i：$p \to B^i{}_P$ 称做二元颗粒化。

定义 6.3[148]　二元粒化结构（Binary Granular Structure）由四元组（V，U，B，C）组成，这里 V 称为对象空间，U 称为数据空间（U，V 可以是相同的集合），$B=\{B^i,\ i=1,2,\cdots,n\}$是一个精确/模糊二元关系的有限集，$C$ 为概念空间的有限集，它由基本邻域 $B^i_P=\{u|u\ \ B^i_P\}$的所有名称（属性值）构成，当 $V=U$ 时二元关系为等价关系，即 $B=E$，这时三元组（U，E，C）称为粗糙颗粒结构（Rough Granular Structure），在该结构中，C 由等价关系和等价类的所有名称组成。

定义 6.4　设 B 为等价关系，由 B 构成了一个等价颗粒化 $B{:}p \to B_p$，由其逆影射 B^{-1} 形成了一个划分，其等价类 $B^{-1}(B_p)$称为基本颗粒 B_P 的颗粒中心。

在信息系统中，设 θ_1，θ_2 为 U 上的两个等价关系，则它们的交定义为：u（$\theta_1 \cap \theta_2$）v，当且仅当 $u\theta_1 v$ 且 $u\theta_2 v$。对于等价关系 θ，对应的划分为 U/θ，对于任意 $u \in U$，有 $u\theta=\{v \in U|v\theta u\}$，这样 $U/\theta=\{u\theta|u \in U\}$，其交定义为 $u^{\theta_1 \cap \theta_2}=u^{\theta_1} \cap u^{\theta_2}$。

当$\theta_1 \cap \theta_2 = \theta_1$，$\theta_1$包含在$\theta_2$中，记为$\theta_1 \subseteq \theta_2$。

把恒等关系记为ε：$u \varepsilon v$，当且仅当对于 u、$v \in U$，有 $u=v$，对于恒等关系有 $U/\varepsilon=\{\{u\}|u \in U\}$，且$|U/\varepsilon|=|U|$。

引理 6.1[40] 设<U，A>为信息系统，那么对于 X，$Y \subseteq A$，有 $\theta_X \cap \theta_Y = \theta_X \ \theta_Y$。

命题 6.1 对于信息系统来说，其每个属性为一个等价关系，每个属性值相当于一个等价类，属性值的集合形成论域的一个划分。

证明：在信息系统<U，A>中，设 B 为属性集 A 的非空子集，根据粗集等价关系原理有：IND（B）=∩{IND（A_i）：$A_i \in B$}，即 IND（B）为所有可能的等价类 IND（A_i）的交集，从这里可以看出，属性子集的每个属性对应着一个等价关系，而 B 为这些等价关系的交集；从上面的面向数据挖掘的机器模型可以看出，每个属性值对应着一个等价颗粒，而在粗集原理中，划分为所有等价类的集合，故每个属性值相当于一个等价类。

由引理 6.1 及上面的分析，可以得出下列结论：

命题 6.2 设{BIT（1），…，BIT（K）}为某个属性对应的等价关系形成的 U 的一个划分，则有 BIT（1）∨…∨BIT（K）=11…1，其中 1 的个数为$|U|$（下同，∨为或操作，∧为与操作）。

证明：设 U/R={BIT（1），…，BIT（K）}为某个属性对应的 U 的一个划分，根据粗集原理有对于任意的 BIT（i）、{BIT（j）$\in U/R$，BIT（i）∧{BIT（j）=0，BIT（1）∨BIT（2）∨…∨BIT（i）∨…∨BIT（K）=BIT(U)，而 BIT(U)=11…1($|U|$个 1)，因此有 BIT（1）∨BIT（2）∨…∨BIT（i）∨…∨BIT（K）=11…1($|U|$个 1)，命题得证。

命题 6.3 设{BIT_s（1），…，BIT_s（K）}和{BIT_t（1），…，BIT_t（L）}为 U 的两个等价划分，则有{BIT_s（1）∨…∨BIT_s（K）}∧{ BIT_t（1）∨…∨BIT_t（L）}=1…1 成立。

证明：设{BIT_s（1），…，BIT_s（K）}和{BIT_t（1），…，

BIT_t（L）}为 U 的两个等价划分，根据命题 6.2 则有：BIT_s（1）$\vee\cdots\vee BIT_s$（K）=11…1(|U|个 1)，BIT_t（1）$\vee\cdots\vee BIT_t$（L）=11…1(|U|个 1)，而显然 11…1(|U|个 1)$\wedge$ 11…1(|U|个 1)= 11…1(|U|个 1)，因此{BIT_s（1）$\vee\cdots\vee BIT_s$（K）}$\wedge${ BIT_t（1）$\vee\cdots\vee BIT_t$（L）}= 1…1 成立。

命题 6.4　设{$BIT_1(A_i(x_1))$, …, $BIT|\theta_{A_i}|(A_i(x_{|\theta_{A_i}|}))$}为 U 的任一个划分，i=1, 2, …, m，则有 $\{BIT_1(A_1(x_1))\vee\cdots\vee BIT|\theta_{A1}|(A_1(x_{|\theta_{A1}|}))\}\wedge\cdots\wedge\{BIT_1(A_m(x_1))\vee\cdots\vee BIT|\theta_{Am}|(A_m(x_{|\theta_{Am}|}))\}$= 1…1 成立。

证明：根据命题 6.2 的结论，对应于属性 A_i 的一个划分 $\{BIT_1(A_i(x_1))$，…，$BIT|\theta_{A_i}|(A_i(x_{|\theta_{A_i}|}))\}$，有 $BIT_1(A_i(x_1))\vee\ldots\vee BIT|\theta_{A_i}|(A_i(x_{|\theta_{A_i}|}))$=11…1(|$U$|个 1)，同理对于 $A_j\in A$ 来说有 $BIT_1(A_j(x_1))\vee\cdots\vee BIT|\theta_{A_j}|(A_i(x_{|\theta_{A_j}|}))$= 11…1(|$U$|个 1)，故对于任意的 A_i，$A_j\in A$，$\{BIT_1(A_i(x_1))\vee\cdots\vee BIT|\theta_{A_i}|(A_i(x_{|\theta_{A_i}|}))\}\wedge\{BIT_1(A_j(x_1))\vee\cdots\vee BIT|\theta_{A_j}|(A_i(x_{|\theta_{A_j}|}))\}$= 11…1(|$U$|个 1)，由此可以归纳得出命题成立。

定义 6.5[150]　两个属性 A_i 和 A_j 是同构的，记为 $A_i\approx A_j$，当且仅当存在一个一对一的映射 s：$Dom(A_i)\to Dom(A_j)$使得对于所有的 $u\in U$，有 $A_i(u)=s(A_j(u))$。

事实上，对于信息系统同构的两列属性来说，一列的属性值能够通过改名变为另一列，而性质不变。

6.4　信息系统函数依赖的信息颗粒原理与计算

6.4.1　函数依赖的信息颗粒原理与计算

函数依赖是关系数据库的核心概念，而信息系统是泛化的数

据库关系，其函数依赖的含义与关系数据库函数依赖相同。

引理 6.2[41] 函数依赖 $X \to Y$ 可以表示为：$\theta_X \subseteq \theta_Y$，即 $\cap_{a\in X}\ \theta_a \subseteq \cap_{a\in Y}\ \theta_a$，这里是$\theta_X$，$\theta_Y$是 U 上的两个等价关系。

设属性（集）X 对应的某个属性值的位表示为 BIT（$v_i(X)$），属性（集）Y 对应的某个属性值的位表示为 BIT（$v_i(Y)$），从信息颗粒的角度讲，对于 X，$Y\subseteq A$，θ_X 用等价颗粒的位表示可以表示为$\{ A_X^{-1}(p)| A_X^{-1}(p)\in \mathrm{BIT}(U)\}\}$，$\theta_Y$ 可以表示为$\{A^{-1}{}_Y(p)|A^{-1}{}_Y(p)\in \mathrm{BIT}(U)\}$，其中 $A_X^{-1}(p)$、$A_Y^{-1}(p)$为信息表某一属性值或子元组的位表示，BIT(U)为论域的位表示，那么对于其函数依赖关系来说，有以下结论：

命题 6.5 同构的两个属性或属性子集必是函数依赖的。

证明：设属性 A_i，A_j 是同构的，对于任意的 u，$v\in U$，若 BIT（A_i，u）=BIT（A_i，v）成立，则必有 BIT（A_j，u）=BIT（A_j，v）成立，即若 $A_i(u)=A_i(v)$成立，则有 $A_j(u)= A_j(v)$成立，满足函数依赖的定义，同理对于属性子集来说也满足函数依赖的定义，命题得证。

定理 6.1 设 X，$Y\subseteq A$，属性（集）X 对应的某个属性值的位表示为 BIT（$v_i(X)$），属性（集）Y 对应的某个属性值的位表示为 BIT（$v_j(Y)$），函数依赖 $X\to Y$ 成立当且仅当：对于任意的 BIT（$v_i(X)$）$\in\{ A_X^{-1}(p)| A_X^{-1}(p)\in \mathrm{BIT}(2^U)\}$，则必有一个 BIT（$v_j(Y)$）$\in$ $\{ A_Y^{-1}(p)|A^{-1}{}_Y(p)\in \mathrm{BIT}(2^U)\}$，使得 BIT（$v_i(X)$）$\wedge$ BIT（$v_j(Y)$）= BIT（$v_i(X)$）成立。

证明：根据引理 6.2，函数依赖 $X\to Y$ 可以表示为：$\theta_X \subseteq \theta_Y$，即 $\cap_{a\in X}\ \theta_a \subseteq \cap_{a\in Y}\ \theta_a$，这里是$\theta_X$，$\theta_Y$ 是 U 上的两个价关系；从 $\theta_X \subseteq \theta_Y$ 可知，对于任意的的子集 $A_i\subseteq\theta_X$，必有一子集 $A_j\subseteq\theta_Y$，使得 $A_i\subseteq A_j$，根据位表示原理，设属性（集）X 对应的某个属性值的位表示为 BIT（$v_i(X)$），属性（集）Y 对应的某个属性值的位表示为 BIT（$v_j(Y)$），则θ_X =$\{ A_X^{-1}(p)| A_X^{-1}(p)\in \mathrm{BIT}(2^U)\}$，

$\theta_Y=\{A_Y^{-1}(p)|A_Y^{-1}(p)\in \mathrm{BIT}(2^U)\}$；设 BIT（$v_i(X)$）$\in\{A_X^{-1}(p)|A_X^{-1}(p)\in \mathrm{BIT}(2^U)\}$，且 BIT（$v_i(X)$）=BIT（$A_i$），BIT（$v_j(Y)$）$\in\{A_Y^{-1}(p)|A_Y^{-1}(p)\in \mathrm{BIT}(U)\}$，且 BIT（$v_j(Y)$）=BIT（$A_j$），由于 $A_i\subseteq A_j$，所以 BIT（$v_i(X)$）∧BIT（$v_j(Y)$）=BIT（$v_i(X)$）成立。

反之，若对于任意的 BIT（$v_i(X)$）$\in\{A_X^{-1}(p)|A_X^{-1}(p)\in \mathrm{BIT}(2^U)\}$，必有一 BIT（$v_j(Y)$）$\in\{A_Y^{-1}(p)|A^{-1}{}_Y(p)\in \mathrm{BIT}(U)\}$，使得 BIT（$v_i(X)$）∧BIT（$v_j(Y)$）=BIT（$v_i(X)$）成立，设 $x_i\in U$，则对于任意的 $x_i\in A_i$，$A_i\subseteq\theta_X$，则必有一 $A_j\subseteq\theta_Y$，使得 $x_i\in A_j$，因此有 $A_i\subseteq A_j$，故 $\theta_X\subseteq\theta_Y$，根据引理 6.2 有函数依赖 $X\to Y$ 成立。定理得证。

表 6-3 是由文献[41]中的例子略加改动而来的：对于属性 A_1 和 A_6，有 θ_{A_1}={{4}，{5}，{3}}={10101，01000，00010}，θ_{A_6}={M，F}={11101，00010}，其中 BIT（4）∧BIT（M）=10101∧11101=10101=BIT（4），BIT（5）∧BIT（M）=01000∧1110101000=BIT（5），BIT（3）∧BIT（F）=00010∧00010= 00010=BIT（3），因此有 $A_1\to A_6$；按照同样的方法，有 $A_2\to A_4$ 成立。

表 6-3　脑（*skull*）信息系统

U	A_1	A_2	A_3	A_4	A_5	A_6
V_1	4	1	*K*	*X*	8.36	*M*
V_2	5	2	*J*	*Y*	5.14	*M*
V_3	4	1	*L*	*X*	8.38	*M*
V_4	3	1	*K*	*X*	8.29	*F*
V_5	4	2	*J*	*Y*	5.27	*M*

6.4.2　恒等依赖的信息颗粒原理与计算

定义 6.6（恒等依赖）信息系统 $I=<U, A>$中，任意两个属性子集 X，$Y\subseteq A$ 之间的恒等依赖描述为：$X\leftrightarrow Y$，它在信息系统中

成立当且仅当 $X \to Y$ 且 $Y \to X$。

引理 6.3[41] 恒等依赖 $X \leftrightarrow Y$ 可以描述 $\theta_X = \theta_Y$，即 $\cap_{a \in X}\ \theta_a = \cap_{a \in Y}\ \theta_a$。

定理 6.2 设 X，$Y \subseteq A$，属性（集）X 对应的某个属性值的位表示为 $\mathrm{BIT}(v_i(X))$，属性（集）Y 对应的某个属性值的位表示为 $\mathrm{BIT}(v_j(Y))$，X、Y 之间的恒等函数依赖 $X \leftrightarrow Y$ 成立，当且仅当对于任意的 $\mathrm{BIT}(v_i(X)) \in \{ A_X^{-1}(p) | A^{-1}{}_X(p) \in \mathrm{BIT}(2^{\mathrm{U}})\}$，则必有一 $\mathrm{BIT}(v_j(Y)) \in \{ A_Y^{-1}(p) | A_Y^{-1}(p) \in \mathrm{BIT}(2^{\mathrm{U}})\}$，使得 $\mathrm{BIT}(v_i(X))=\mathrm{BIT}(v_j(Y))$ 成立。

证明：根据引理 6.3，恒等依赖 $X \leftrightarrow Y$ 可以描述 $\theta_X = \theta_Y$，即 $\cap_{a \in X}\ \theta_a = \cap_{a \in Y}\ \theta_a$。这里是 θ_X，θ_Y 是 U 上的两个等价关系；从 $\theta_X = \theta_Y$ 可知，对于任意的的子集 $A_i \subseteq \theta_X$，必有一子集 $A_j \subseteq \theta_Y$，使得 $A_i=A_j$，根据位表示原理，设属性（集）X 对应的某个属性值的位表示为 BIT（$v_i(X)$），属性（集）Y 对应的某个属性值的位表示为 BIT（$v_j(Y)$），则 $\theta_X = \{ A_X^{-1}(p) | A_X^{-1}(p) \in \mathrm{BIT}(U)\}$，$\theta_Y = \{ A_Y^{-1}(p) | A_Y^{-1}(p) \in \mathrm{BIT}(U)\}$；设 $\mathrm{BIT}(v_i(X)) \in \{A^{-1}{}_X(p) | A_X^{-1}(p) \in \mathrm{BIT}(U)\}$，且 $\mathrm{BIT}(v_i(X))=\mathrm{BIT}(A_i)$，BIT（$v_j(Y)$）$\in \{A^{-1}{}_Y(p) | A_Y^{-1}(p) \in \mathrm{BIT}(U)\}$，且 $\mathrm{BIT}(v_j(Y))=\mathrm{BIT}(A_j)$，由于 $A_i=A_j$，所以 $\mathrm{BIT}(v_i(X))=\mathrm{BIT}(v_j(Y))$成立。

反之，若对于任意的 $\mathrm{BIT}(v_i(X)) \in \{ A_X^{-1}(p) | A_X^{-1}(p) \in \mathrm{BIT}(U)\}$，必有一个 $\mathrm{BIT}(v_j(Y)) \in \{ A_Y^{-1}(p) | A_Y^{-1}(p) \in \mathrm{BIT}(U)\}$，使得 $\mathrm{BIT}(v_i(X))=\mathrm{BIT}(v_j(Y))$成立，设 $x_i \in U$，则对于任意的 $x_i \in A_i$，$A_i \subseteq \theta_X$，则必有一 $A_j \subseteq \theta_Y$，使得 x_i，$\in A_j$，因此有 $A_i=A_j$，故 $\theta_X = \theta_Y$，根据引理 6.3 有恒等依赖 $X \leftrightarrow Y$，定理得证。

对于恒等依赖来说，并不是其对应的属性值相等，而是对应的属性值内部表示——位表示相同。

命题 6.6 同构的两个属性或属性子集必是恒等依赖的。

证明：恒等依赖为函数依赖的特殊情况，按照函数依赖的命题 6.5 本命题同理成立。

命题 6.7　对于恒等依赖 $X \leftrightarrow Y$，即 $\cap_{a\in X}\ \theta_a = \cap_{a\in Y}\ \theta_a$，有 $|\theta_X| = |\theta_Y|$成立。

证明：本命题是显然的，对于恒等依赖的两个属性或属性集来说，由 $\cap_{a\in X}\ \theta_a = \cap_{a\in Y}\ \theta_a$ 成立可以看出两个等价关系的分类相同，即 $card(\theta_X)=card(\theta_Y)$，意味着$|\theta_X| = |\theta_Y|$成立。

表 6-3 中，对于 A_2，A_4，属性值 1 与 X 的位表示相同，属性值 2 与 Y 的位表示相同，因此有恒等依赖 $A_2 \leftrightarrow A_4$ 成立。

显然，恒等依赖是函数依赖的特殊情况。

6.4.3　部分依赖的信息颗粒原理与计算

定义 6.7[126]：设 $I=<U, A>$为信息系统，P，$Q \subseteq A$ 为属性子集，属性集 Q 依赖于属性集 P 的程度为：$K=\gamma_P(Q)=card(\mathrm{POS_p}(Q))/card(U)$，其中，$POS_p(Q)$为 Q 的 P 正域。当 $K=1$ 时，I 中属性集 Q 全依赖于属性集 P，当 $0<K<1$ 时，I 中属性集 Q 部分依赖于属性集 P，当 $K=0$ 时，属性集 Q 独立于属性集 P。

定理 6.3　设 X、$Y \subseteq A$，属性（集）X 对应的某个属性值的位表示为 $\mathrm{BIT}(v_i(X))$，属性（集）Y 对应的某个属性值的位表示为 $\mathrm{BIT}(v_j(Y))$，部分函数依赖 $X \xrightarrow{p} Y$ 成立当且仅当存在 $\mathrm{BIT}(v_i(X)) \in \{ A_X^{-1}(p) | A_X^{-1}(p) \in \mathrm{BIT}(2^U)\}$，有 $\mathrm{BIT}(v_j(Y)) \in \{A^{-1}{}_Y(p) | A_Y^{-1}(p) \in \mathrm{BIT}(2^U)\}$，使得 $\mathrm{BIT}(v_i(X)) \wedge \mathrm{BIT}(v_j(Y)) = \mathrm{BIT}(v_i(X))$成立。

证明：根据部分依赖的定义仿定理 6.1 与 6.2 可以证得本定理成立。

例：对于一信息系统来说，有相应的分类如下：$U=\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$, $P=\{Y_1, Y_2, Y_3, Y_4, Y_5, Y_6\}$, $Q=\{X_1, X_2, X_3, X_4, X_5\}$, $U/Q=\{10000000, 01000010, 00100100, 00010000, 00001001\}$, $U/P=\{10001000, 01000001, 00100000, 00010000, 00000100, 00000010\}$, 对于 U/Q 来说，有 $00000010 \in U/P$，且 $P_{-}(X_2)=00000010 \wedge 01000010=00000010$，$P_{-}(X_3)=\{00100000, 00000100\}$，$P_{-}(X_4)=\{00010000\}$，

因而 $P \xrightarrow{p} Q$ 成立，且 $POS_p(Q)$={00000010，00100000，00000100，00010000}，$r_p(Q) = 4/8 = 0.5$。

6.5 算法描述与分析

下面给出利用位表示法来检测两个属性或属性集 X，$Y \subseteq A$ 之间是否存在函数依赖 $X \to Y$ 的算法。

算法：判定信息系统中任意两个属性或属性集之间的函数依赖关系。

设<U，A>为一个信息系统，X，$Y \subseteq A$。

利用信息颗粒位操作方法求出

（1）U/θ_X ={Bit(X_{11})，Bit(X_{12})，…，Bit(X_{1i})，…，Bit($X_{|n1|}$)}，$|n1| \leqslant |U|$，U/θ_Y ={Bit(Y_{21})，Bit(Y_{22})，…，Bit(Y_{2j})，…，Bit($Y_{|n2|}$)}，$|n2| \leqslant |U|$

（2）[初始化]. Set $i \leftarrow 1$，$j \leftarrow 1$

（3）[Bit(X_{1i})∧ Bit(Y_{2j})= Bit(X_{1i})?]

（4）*if* Bit(X_{1i})∧(BitY_{2j})=(BitX_{1i}) *then*

（5）（已发现一个 j 使得 Bit(X_{1i})∧(BitY_{2j})= Bit(X_{1i}) 成立）转（7）检查下一个 i

（6）If Bit(X_{1i})∧Bit(Y_{2j})≠ Bit(X_{1i}) then goto (10) 检查下一个 j

（7）[i=|$n1$|?] if i=|$n1$|then 算法完成检测 $X \to Y$，函数依赖成立

（8）if i<|$n1$| then goto (9)

（9）[Increase i]. Set $i \leftarrow i$+1，go to (3)

（10）[j=|$n2$|?] if j=|$n2$|(存在一个 i 使得 Bit(X_{1i})∧Bit(Y_{2j})≠Bit(X_{1i}) 对于 j=1，2，…，|$n2$|)then 算法完成：$X \to Y$ 不成立

（11）If j<|$n2$| then goto (12)

（12）[Increase j]. Set $j \leftarrow j$+1，goto (3)

该算法的时间复杂度为 O（$|U|^2$）。

按照判定函数依赖的算法可以类似地给出判定恒等依赖与部分依赖的算法，其算法的时间复杂度也为 $O(|U|^2)$。

算法举例：

如在表 6-1 中，输入 X 为 Sname，Y 为 Status，则：U/θ_X ={10000，01000，00100，00010，00001}，U/θ_Y ={10010，01100，00001}，为根据本文的方法可以得出下列函数依赖：

Sname→Status 成立。

输入 X 为 Sname，Y 为 City，则：U/θ_X ={10000，01000，00100，00010 00001}　U/θ_Y ={10010，01100，00001}，为根据本文的方法可以得出下列函数依赖：

Sname→City 成立.

输入 X 为 Status，Y 为 City，则：U/θ_X ={10010，01100，00001}，U/θ_Y ={10010，01100，00001}，为根据本文的方法可以得出下列函数依赖：

S4atus→City，Status↔City 成立。

函数依赖的判定对于知识发现是一个很重要的概念，它可以以一种很简化的形式说明给定的信息系统中哪些值是有效的。利用发现的函数依赖，可以把它们用于数据的约简、规则一致性的判定和查询优化等。

传统的关系数据库一般用语义来判定函数依赖，粗糙集利用不可分辨原理及相应的度量公式来判定数据库属性间的函数依赖性，本文采用信息颗粒的位表示方法研究了信息系统函数依赖的判定问题，根据本方法可以发现数据库潜在的未知的函数依赖，本文方法丰富和扩充对函数依赖的研究思路。与粗糙集方法相比较，本方法具有以下特点：

（1）本方法以粗糙集的等价关系和信息颗粒原理为基础，可以迅速判定和度量信息系统或数据库的属性间的函数依赖关系：对于要判定的两个属性列来说，每个属性值只需要做很少次数的判断（不超过 *card* (ind (*Y*))），而不需要像传统的方法那样做多次

判定，因此提高了判定的效率。

（2）从数据处理方式上说，粗糙集方法对于等价颗粒采用集合表示，而本文对于属性值采用位表示，相比较更接近机器的内部表示，运算效率与速度得到了提高。

（3）对于关系数据库来说，其现实的数学结构是粗糙粒化结构，因此本方法能够直接挖掘信息系统或数据库所蕴含的各种模式，如函数依赖关系、关联规则和决策规则等。

本章利用粗集等价关系原理和信息颗粒的位表示方法给出了信息系统任意两个属性间的普通函数依赖、恒等依赖和部分依赖的判定方法，拓宽了对于函数依赖的研究，进一步可以研究利用位方式判定数据库多值依赖，求其函数依赖集等。

第 7 章　关系数据库范式及信息系统规则的研究

为学患无疑，疑则进也
——陆九渊《陆九渊集》

7.1　引言

众所周知，关系数据库是以关系模型为基础的，它利用关系描述现实世界。数据依赖是通过一个关系中属性值的相等与否体现出来的数据间的相互关系，是现实世界属性间相互联系的抽象，是语义的体现，有许多类型的数据依赖，其中最重要的是函数依赖和多值依赖。

函数依赖是关系数据库最重要的概念之一。数据库的属性间往往存在一定的依赖关系，而最基本的依赖关系是函数依赖，所谓函数依赖是指一个关系中一个或一组属性的值能够决定其他属性的值。函数依赖一般用于数据库的逻辑设计，用以表示完整性约束。

关系数据库模式规范化是关系数据库研究的一个重要领域，如果只考虑函数依赖，则属于 BCNF 的关系模式规范化程度已经是最高的了。

在关系模型中，有两个重要概念：范式与函数依赖，它们是关系规范化理论的核心。关系规范化理论是关系型数据库逻辑设计的基础，系统开发人员对关系规范化的运用能力将直接影响所设计数据库系统的质量，并进而影响整个信息系统的性能。范式

的概念最早是由 IBM 公司的研究员 E. F. Codd 提出的，他于 1971—1972 年发表系列论文系统地提出了 1NF，2NF 和 3NF 的标准，并深入探讨了关系进一步规范化的问题，由此奠定了关系规范化理论的基础，1974 年，E. F. Codd 和 Boyce 又共同提出了 BCNF[151]。

虽然关系规范化的理论研究发展至今已经比较完备，但仍有进一步完善和充实的必要，这是由于在对于关系数据库函数依赖（关系规范化理论）的判定上，传统的方法主要是利用语义进行的，在具体运用上可操作性一般，而粗糙集的产生为关系规范化理论注入了新的活力，使我们可以以一种新的视角来研究关系规范化理论。本书便是利用粗糙集理论和方法来研究关系数据库的关系范式的判定问题，主要解决给定任意一个关系后，判定其隶属哪个范式的问题。

7.2 函数依赖与范式

定义 7.1 设 $R=\{A_1, A_2, \cdots, A_n\}$为具有 n 个属性的关系模式，X，$Y \subseteq A$，X，Y 之间的函数依赖记为 $X \rightarrow Y$，其依赖性描述为当 $t_i[X]=t_j[X]$时，必有 $t_i[Y]=t_j[Y]$，其中 t_i、t_j 为数据库的两个元组，称 X 函数决定 Y，或 Y 函数依赖于 X。

对于函数依赖，需要说明以下几点：

函数依赖不是指关系模式 R 的某个或某些关系实例满足的约束条件，而是指 R 的所有关系实例均要满足的约束条件。

函数依赖和别的数据之间的依赖性一样，是语义范畴的概念，只能根据数据的语义来确定函数依赖。

数据库设计者可以对现实世界做强制的规定，即指定相应的函数依赖。

定义 7.2[124] 对于满足一组函数依赖 F 的关系模式 $R<U, F>$，其任意一个关系 r，若函数依赖 $X \rightarrow Y$ 都成立，（即 r 中任意两元

组 t，s，若 $t[X]=s[X]$，则 $t[Y]=s[Y]$），则称 F 逻辑蕴含 $X\rightarrow Y$。

为了从一组函数依赖求得蕴含的函数依赖，如已知函数依赖集 F，要问 $X\rightarrow Y$ 是否为 F 所蕴含，就需要一套推理规则，这组推理规则是 1974 年首先由 Armstrong 提出来的。

Armstrong 公理系统　设 U 为属性集总体，F 是 U 上的一组函数依赖，于是有关系模式 $R<U, F>$，对 $R<U, F>$来说有以下的推理规则：

（1）自反律（Reflexivity）若 $Y\subseteq X\subseteq U$，则 $X\rightarrow Y$ 为 F 所蕴含。

（2）增广律（Augmentation）若 $X\rightarrow Y$ 为 F 所蕴含，且 $Z\subseteq U$，则 XZ→YZ 为 F 所蕴含。

（3）传递律（Transitivity）若 $X\rightarrow Y$ 及 $Y\rightarrow Z$ 为 F 所蕴含，则 $X\rightarrow Z$ 为 F 所蕴含。

Armstrong 公理的有效性指由 F 出发根据 Armstrong 公理推导出来的每一个函数依赖一定在 F^+中，F^+为 F 的闭包，即 F 所蕴含的函数依赖的全体。

Armstrong 公理的完备性指的是：F^+中的每一个函数依赖，必定可以由 F 出发根据 Armstrong 公理推导出来。

引理 7.1[124]　Armstrong 推理规则是正确的。

引理 7.2[124]　Armstrong 公理系统是有效的和完备的。

定义 7.3[152]　关系模式是对关系的描述，它是一个四元组：$R<U, D$，DOM，$F>$，在关系模式中，影响数据库模式设计的主要是 U 和 F，因此通常把它简记为一个三元组 $R<U, F>$，式中，R 为关系名，U 为组成该关系的属性名的集合，D 为属性组 U 中属性所来自的域，DOM 为属性向域的映像的集合，F 为属性间数据依赖关系的集合。

函数依赖是关系模式中一部分属性的值依赖于另一部分属性的值这样一种依赖关系，它是 $R<U, F>$的一切关系均要满足的约束条件。

定义 7.4[124]　在 $R<U, F>$中，如果 $X\rightarrow Y$，并且对于 X 的任

何一个真子集 X'，都有 $X'\rightarrow Y$，则称 Y 对 X 完全函数依赖，记作：$X\xrightarrow{F}Y$；若 $X\rightarrow Y$，但 Y 不完全依赖于 X，则称 Y 对 X 部分函数依赖。

定义 7.5 在关系模式 $R<U，F>$的每个关系中，如果每个属性值都是不可再分的原子值，那么 R 是满足第一范式的模式，记作 $R\in$1NF。

定义 7.6 如果关系模式 $R<U，F>$满足 1NF，且每个非主属性完全函数依赖于候选键，那么 $R<U，F>$是满足第二范式的模式，简记为 $R\in$2NF。

定义 7.7 如果关系模式 $R<U，F>$满足 1NF，且每个非主属性不传递依赖于 $R<U，F>$的某个候选键，则 $R<U，F>$为满足第三范式的模式，简记为 $R\in$3NF。

定义 7.8 如果关系模式 $R<U，F>$满足 1NF，且每个属性都不传递函数依赖于 $R<U，F>$的候选键，则称 $R<U，F>$满足 $R\in$BCNF。

在粗糙集理论中，设 U 是非空有限论域，R 是 U 上的二元等价关系，R 称为不可分辨关系，序对 $A=(U，R)$称为近似空间。$\forall(x，y)\in U\times U$，若$(x，y)\in R$，则称对象 x 与 y 在近似空间 A 中是不可分辨的。U/R 是 U 上由 R 生成的等价类全体，它构成了 U 的一个划分[5]。

粗糙集理论的知识表达方式一般采用信息表或称为信息系统的形式，它可以表示为四元有序组 $K=(U，A，V，\rho)$，其中 U 是对象的全体，即论域；A 是属性全体；$V=\bigcup_{a\in A}V_a$；V_a 是属性的值域；ρ：$U\times A\rightarrow V$ 是一个信息函数，ρ_x：$A\rightarrow V$，$x\in U$，反映了对象 x 在 K 中的完全信息，其中$\rho_x(a)=\rho(x,a)$。

对于这样的信息系统，每个属性子集就定义了论域上的一个等价关系，即$\forall B\subseteq A$，定义 R_B：$xR_By\Leftrightarrow\rho_x(b)=\rho_y(b)$，$\forall b\in B$。

前面给出了函数依赖、划分和等价关系的基本定义，按照粗

糙集原理，划分、等价关系及关系数据库的函数依赖之间是有一定关系的，简要分析如下。

定义非空有限论域 U 上的偏序称为细分（Refinement）[153]，设 P_1，P_2 为 U 的两个划分。如果对于任意的 $S_1 \in P_1$，有 $S_1 \subseteq S_2$，$S_2 \in P_2$，则称 P_1 是 P_2 的细分，记为 $P_1 \subseteq P_2$。

下面给出函数依赖的另外一个等价的形式化描述：设 R 是一个数据库模式，α，$\beta \subseteq R$，$\alpha \rightarrow \beta$ 是 R 上的函数依赖，则有：（$\forall s$，$t \in r$）($s[\alpha]=t[\alpha] \Rightarrow s[\beta]=t[\beta]$)，$r(R)$满足$\alpha \rightarrow \beta$。

引理 7.3[153]　设$\alpha \rightarrow \beta$为关系模式 $R<U$，$F>$上的函数依赖，$r(R)$为 R 的一个关系，那么 $r(R)$满足$\alpha \rightarrow \beta$，当且仅当 $P_\alpha \subseteq P_\beta$ 成立，这里 P_α，P_β为由α，β产生的等价类。

上述引理表明了对于关系数据库的任意一个关系，判定其是否满足给定的函数依赖可以通过划分或等价关系来进行，按照这样的原理，对于关系数据库的任一关系，都可以判定其所满足的范式，即其达到哪个级别的范式。

7.3　基于粗糙集理论的关系模式范式的判定原理

从前面的研究可以看出，划分、等价关系与关系数据库的函数依赖之间是有一定关系的，而范式是用于对关系数据库进行规范化的函数依赖关系，因此可以进一步研究等价关系与关系模式之间的关系，研究给定相应的范式后，关系所满足的性质，以及给定相应的关系后，其所属的范式级别。

在关系规范化理论中，数据库模式是关系模式的集合，对于任一数据库模式有：

定理 7.1　设 $R<U$，$F>$为关系数据库模式，$r \in R<U$，$F>$为 $R<U$，$F>$的任一关系，那么 $r \in$1NF，当且仅当$|t[A]|=1$，这里$|t[A]|$表示属性值 $t[A]$所包含的元素的个数。

上述定理是显然的，某个关系是否满足 1NF 关键是看其每一

个属性值是否为不可再分的原子值，若不可再分，其为原子值，其包含的元素个数为 1，满足 1NF 的定义，若可分，则至少有一属性值 $t[A]$，使得$|t[A]|>1$；反之，若某个关系满足 1NF，则其每个属性值都是原子的，对任意的 $t[A]\in r$，有$|t[A]|=1$。

关系模式 R =<{E#，JC，D#，M#，CT}>，其中，E#：employee number；JC：job code；D#：department number；M#：employee number of manager；CT：contract type。其构成的关系如表 7-1 所示（根据文献[154]修改）：

表 7-1　职工信息

E#	JC	D#	M#	CT
1	A	x	11	g
2	C	x	11	g
3	A	y	12	n
4	B	x	11	g
5	B	y	12	n
6	C	y	12	n
7	A	z	13	n
8	C	z	13	n

根据定理 7.1，表 7-1 满足 1NF，因为它的每个属性值 $t[A]$都是原子的，即$|t[A]|=1$。

定理 7.2　设 $R<U，F>$为关系数据库模式，$r\in R<U，F>$为 $R<U，F>$的任一关系模式，设 r 的主属性集为 $Z=\{key_1，key_2，\cdots，key_n\}$，非主属性集为 Z'，若 $r\in$2NF，则对于任意的 $attr\in Z'$，均有 $P_{key_i}\subseteq P_{attr}$成立且 $P_k\subseteq P_{attr}$不成立，其中 k 为 key_i的真子集，P 为前面所定义的划分。

证明：设 $R<U，F>$为任意的关系数据库模式，r 为其关系模式，该关系模式的主属性集为 $Z=\{key_1，key_2，\cdots，key_n\}$，非主属

性集为 Z'，现已知 $r\in$2NF，那么 r 的非主属性对其任意的候选码则是完全依赖，而不存在部分依赖，那么对任意的 $attr\in Z'$，有 $key_i\rightarrow attr$ 与 $k\rightarrow attr$ 成立，其中 k 为 key_i 的真子集，按照定理 7.1 的结论，则有 $P_{key_i}\subseteq P_{attr}$ 成立且 $P_k\subseteq P_{attr}$ 不成立，本定理得证。

在表 7.1 中，该关系模式满足 2NF，其主属性集为{E#}，非主属性集为{JC，D#，M#，CT}，因而 JC，D#，M#，CT 对 E# 有 U/E#$\subseteq U$/JC(U/E#即 $P_{E\#}$，下同)、U/E#$\subseteq$U/D#、U/E#$\subseteq U$/M#、U/E#$\subseteq U$/CT 成立，而 E#无真子集（ϕ除外）。在由 E#形成的等价划分 U/E#={{1}，{2}，{3}，{4}，{5}，{6}，{7}，{8}}，U/JC={{1，3，7}，{2，6，8}，{4，5}}中，显然有 U/E#$\subseteq U$/JC 成立，依次类推，有 U/E#$\subseteq U$/D#，U/E#$\subseteq U$/M#，U/E#$\subseteq U$/CT 成立。

与定理 7.2 相反，可以给出任意一个关系是否满足 2NF 的判定定理。

定理 7.3　设 $R<U，F>$为关系数据库模式，$r\in R<U，F>$为 $R<U，F>$的任一关系模式，$r\in$1NF，设 r 的主属性集为 $Z=\{key_1，key_2，\cdots，key_n\}$，非主属性集为 Z'，若$\exists attr\in Z'$，使得 $P_{key_i}\subseteq P_{attr}$ 成立且 $P_k\subseteq P_{attr}$ 成立，那么关系模式 $r\notin$2NF，其中 k 为 key_i 的真子集，P 为前面所定义的划分；反之，若对任意关系模式 $r\in$1NF，$\forall attr\in Z'$，均有 $P_{key_i}\subseteq P_{attr}$ 成立且 $P_k\subseteq P_{attr}$ 不成立，则 $r\in$2NF。

证明：已知 $R<U，F>$为关系数据库模式，$r\in R<U，F>$为 $R<U，F>$的任一关系模式，其主属性集为 $Z=\{key_1，key_2，\cdots，key_n\}$，非主属性集为 Z'，若$\exists attr\in Z'$，使得 $P_{key_i}\subseteq P_{attr}$ 成立且 $P_k\subseteq P_{attr}$ 成立，那么按照定理 7.1 有 $key_i\rightarrow attr$ 且 $k\rightarrow attr$ 成立，即有一非主属性 $attr$ 部分依赖于一候选健 key_i，按照 2NF 的定义，有 $r\notin$2NF。反之，若对任意关系模式 r，对$\forall attr\in Z'$，均有 $P_{key_i}\subseteq P_{attr}$ 成立且 $P_k\subseteq P_{attr}$ 不成立，则有 $key_i\rightarrow attr$ 且 $k\rightarrow attr$ 成立，那么属性 $attr$ 对 key_i 是完全函数依赖，按照 2NF 的定义，有 $r\in$2NF，定理得证。

在表 7-1 中，主属性集为{E#}，非主属性集为 Z'={JC，

D#，M#，CT}，由 E#可以形成等价划分 U/E# ={{1}，{2}，{3}，{4}，{5}，{6}，{7}，{8}}，U/JC={{1，3，7}，{2，6，8}，{4，5}}，其中显然有 U/E#⊆U/JC 成立，由 D#可以形成等价划分 U/D#={{1，2，4}，{2，5，6}，{7，8}}，其中显然有 U/E#⊆U/D#成立，依次类推，U/E#⊆U/M#，U/E#⊆U/CT 成立，而 E#无真子集（ϕ除外），因此可以断定该数据库关系满足 2NF。

定理 7.4 设 $R<U，F>$为关系数据库模式，$r∈R<U，F>$为 $R<U，F>$的任一关系模式，r 的主属性集为 $Z=\{key_1，key_2，…，key_n\}$，非主属性集为 Z'，若 $r∈$3NF，则对于$\forall attr∈Z'$，均有 $P_{key_i}⊆P_{attr}$ 成立，且不存在一属性（组）k 及 key_i 的真子集 kk 使得 $P_{key_i}⊆P_k$、$P_k⊆P_{attr}$ 及 $P_{kk}⊆P_{attr}$ 成立，其中 P 为前面所定义的划分。

上述定理给出了具有 3NF 的关系模式的非主属性所满足的性质，该定理可以仿照定理 7.2 和定理 7.3 证得。那么给出一任意关系，也同样可以按照粗糙集原理判定其是否满足 3NF。

定理 7.5 设 $R<U，F>$为关系数据库模式，$r∈R<U，F>$为 $R<U，F>$的任一关系模式，$r∈$1NF，设 r 的主属性集为 $Z=\{key_1，key_2，…，key_n\}$，非主属性集为 Z'，对于$\forall attr∈Z'$，若$\exists k$ 使得 $P_{key_i}⊆P_k$ 成立且 $P_k⊆P_{attr}$ 成立，那么关系模式 $r∉$3NF，其中 k 为 r 的任意属性（组），P 为前面所定义的划分；反之，若对任意关系模式 $r∈$1NF，对$\forall attr∈Z'$，均有 $P_{key_i}⊆P_{attr}$ 成立，且①不存在 key_i 的真子集 k 使得 $P_k⊆P_{attr}$；②不存在一属性（组）kk 使得 $P_{key_i}⊆P_{kk}$、$P_{kk}⊆P_{attr}$，那么 $r∈$3NF。

仍以表 7-1 为例，主属性集为{E#}，非主属性集为 Z'={JC，D#，M#，CT}，其中至少存在属性 D#，使得 U/E#={{1}，{2}，{3}，{4}，{5}，{6}，{7}，{8}}⊆U/D#={{1，2，4}，{2，5，6}，{7，8}}成立，而由 M#形成的等价划分为 U/M#={{1，2，4}，{2，5，6}，{7，8}}，因此 U/D#⊆U/M#，也就是说在该关

系中存在有传递依赖，因此 $r \notin 3NF$。

定理 7.6　设 $R<U, F>$为关系数据库模式，$r \in R<U, F>$为 $R<U, F>$的任一关系模式，r 的属性集为 $U=\{attr_1, attr_2, \cdots, attr_n\}$，若 $r \in BCNF$，则有①对于$\forall attr_i \in U$（与定理 7.4 不同，那里的为非主属性集），有 $P_{key_j} \subseteq P_{attr_i}$ 成立，其中，P_{key_j}，P_{attr_i} 分别为由 key_j，$attr_i$ 所形成的划分；②且不存在一属性（组）k 及 key_i 的真子集 kk 使得 $P_{key_i} \subseteq P_k$，$P_k \subseteq P_{attr}$ 及 $P_{kk} \subseteq P_{attr}$ 成立，其中，P 为前面所定义的划分。

上述定理描述了满足 BCNF 的数据库模式中的关系模式的所有属性所应满足的性质。

定理 7.7　设 $R<U, F>$为关系数据库模式，$r \in R<U, F>$为模式 $R<U, F>$的任一关系，r 的属性集为 $U=\{attr_1, attr_2, \cdots, attr_n\}$，$r \in 1NF$，若对于$\forall attr_i \in U$，均有 $P_{key_j} \subseteq P_{attr_i}$ 成立，且不存在一属性（组）k 及 key_i 的真子集 kk 使得 $P_{key_i} \subseteq P_k$，$P_k \subseteq P_{attr}$ 及 $P_{kk} \subseteq P_{attr}$ 成立，其中，P 为前面所定义的划分，则 $r \in BCNF$，其中，P_{key_j}，P_{attr_i} 分别为由 key_j，$attr_i$ 所形成的划分。

对于表 7-1，其主属性集为{E#}，非主属性集为 Z'={JC，D#，M#，CT}，按照定理 7.7 进行判定，虽然有 U/E#$\subseteq U$/JC U/E#$\subseteq U$/D#，U/E#$\subseteq U$/M#，U/E#$\subseteq U$/CT 成立，但是如前所验证，其中有部分 E#→D#，D#→*M*#依赖存在，故该关系模式不满足 BCNF。

7.4　信息系统软规则及其度量关系的研究

本节利用信息颗粒的位表示（Bit Representations）来进行信息系统软规则及其度量之间关系的研究。具体地，在给出相关的基本概念如软规则、EFD 和 IFD 的基础上，首先利用软规则对关联规则、决策规则和函数依赖之间的关系进行分析，然后对关联规则度量、决策规则度量和外延的函数依赖度量的关系进行研

究，并且建立这些度量的统一模型。

函数依赖性的研究有很多应用，它们常用于数据库的规范化及设计、查询优化、数据约简和规则提取中。在数据库知识发现系统中，依赖性的分析结果可以直接引起终端用户的兴趣，如揭示未知的属性函数依赖性，一般地，强依赖性反映了数据库固有的域结构。自动地发现检测依赖性可以为模式抽取算法发现知识提供一个有益的方法。

著名美籍华人学者 T.Y.Lin 教授近年来发表系列论文研究了与信息颗粒、颗粒计算有关的关系数据库的面向机器的数据挖掘建模理论问题，提出了利用信息颗粒的位表示来进行数据挖掘的思想，他主要的工作是把该思想用于关系数据库的建模及各种关联规则的发现，其研究方法为数据挖掘提供了一种新的思路。

Lotfi A.Zadeh 教授于 1965 年提出了模糊集理论，并于 1979 年研究了模糊集与信息粒度（Information Granularity）的关系问题。20 世纪 90 年代，T.Y.Lin 教授提出了粒计算方法，Zadeh 给出了粒计算的定义：“GrC 是模糊信息粒化、粗糙集理论和区间计算的超集，是粒化数学的子集[70]”，可以说，粒计算是在问题求解过程中对颗粒使用的一种理论、公理、技术或工具。

本节基于粒计算和面向机器的数据挖掘原理，对于关联规则、决策规则、函数依赖及其度量之间的关系进行了系统的研究。

7.4.1 信息颗粒的位表示

在关系数据库中，由于属性值的机器语义是一些对象的集合，因而可以用表示颗粒的位模式来取代属性值。按照粗集原理，论域被划分为互不相交的且其并集为论域的非空子集，每个非空子集被称为等价类，而等价类的集合称为商集，记为 U/R，每个等价类充当两个角色：商集的一个元素和论域的一个子集；作为商集的一个元素，称之为规范名，它是第二个角色子集的标

识，因此可以说商集是由规范名构成的，每个规范名表示的是一个等价类。

在一个信息系统中，设 U 为论域（对象的集合），C 为基本概念的集合（属性值），m 为 U 到 C 的映射，m: $U\to C$，则（x，$m(x)$）称为信息表，它兼顾对象及其对应的属性值，而关系数据库则只关心属性值本身，即 $m(x)$。

定义 7.9　基本的信息颗粒定义为 $EF_K(u)$，这里 $EF_K(u)$为 $A_i=A_i(u)$ 的选择子的连接，即$\| EF_K(u)\|_{IS}=\| \wedge_{A_i\in K} A_i=A_i(u)\|_{IS}$，其中 $K\subseteq A$，$u\in U$，$\|\cdot\|$为公式集 ϕ 到幂集 2^U 的映射函数[65]。

从本质上讲，“颗粒”即基本元素，信息颗粒是在基本集（粗糙集概念）中具有相同或相似属性值的对象集合，是通过不可分辨性、相似性和函数性等来划分的对象的集合。

如同把计算分为硬计算（Hard Computing）和软计算（Soft Computing）一样，可以把规则分为硬规则（Hard Rules）和软规则（Soft Rules）。

定义 7.10　设 $A=\{A_1, A_2, \cdots, A_n\}$，$B=\{B_1, B_2, \cdots, B_m\}$ 为关系数据库的两个任意属性集，$c=(a_1, a_2, \cdots, a_n)$，$d=(b_1, b_2, \cdots, b_m)$为对应于 A，B 的两个元组，G_{a_i}，G_{b_i} 为对应于 a_i 和 b_i 的两个基本颗粒，$i=1, 2, \cdots, n$，$j=1, 2, \cdots m$，$P_c=\cap_i G_{a_i}$，$P_d=\cap_j G_{b_i}$ 为其分别对应的交，当 $P_c\dot{\subseteq}Q_d$（$\dot{\subseteq}$为软包含）时，$c\to d$ 是一个软规则[155]。

其中 $\dot{\subseteq}$ 可以解释为集合论意义下的普通包含，也可以是按照阈值定义的包含。

硬规则可以形式化地表示为：if　A　then　B，软规则可以形式化地表示为：if　A　then　probably B，利用软规则来统一不同的规则。

7.4.2 几种规则及其度量之间的关系

在信息系统中，规则可以用各种形式表示，一般包含关联规则、决策规则和外延的函数依赖，它们分别对应着不同的度量，下面研究这些规则及其度量之间的关系。

关联规则发现的主要对象是事务数据库，设 $R=\{I_1, I_2, \cdots, I_m\}$是一组物品集，$W$ 是一组事务集，W 中的每个事务 T 是一组物品，$T \subset R$。假设有一个物品集 A，一个事务 T，如果 $A \subset T$，则称事务 T 支持物品集 A。关联规则是如下形式的一种蕴含：$A \rightarrow B$，其中，A，B 是两组物品，$A \subset I$，$B \subset I$，且 $A \cap B = \phi$。常用下面两种度量来描述关联规则的属性。

定义 7.11[1] （关联规则的支持度）设 W 中有 S%的事务同时支持物品集 A 和 B，S%称为关联规则 $A \rightarrow B$ 的支持度，可表示为 Support($A \rightarrow B$)=Pr($A \cup B$)。

定义 7.12[1] （关联关则的置信度）设 W 中物品集 A 的事务中，有 C%的事务同时也支持物品集 B，C%称为关联规则 $A \rightarrow B$ 的置信度，可表示为 Confidence($A \rightarrow B$)=Pr($B|A$)。

决策表属性一般分为两类：条件属性和决策属性，决策表的每一行可以解释为一条决策规则，按照其度量性质可以把规则分为一致性规则（确定的、非冲突的）和不一致规则（非确定的、冲突的），有时也称为确定性规则和可能性规则。

定义 7.13 （属性的依赖性度量）[126] 设 C、D 为属性集 A 的属性子集，C 为条件属性，D 为决策属性，则将 D 为 K（$0 \leqslant K \leqslant 1$）度依赖于 C，记为 $C \Rightarrow_K D$，$K=\gamma(C, D) = \dfrac{|\mathrm{POS}_C(D)|}{|U|}$，这里 $\mathrm{POS}_C(D)=\mathrm{POS}_C(d_D)$ 很明显，$\gamma(C, D) = \sum\limits_{X \in U/D} \dfrac{|\underline{C}(X)|}{|U|}$，如果 $K=1$，则 D 全依赖于 C，如果 $0<K<1$，则 D 部分依赖于 C，如果 $K=0$，则 D 不依赖于 C。

决策表规则中所有的一致性规则的数量可用于决策表的一致性度量，记为$\gamma(C, D)$，如果$\gamma(C, D)=1$，那么决策表是一致的，如果$\gamma(C, D)\neq 1$，那么决策表是不一致的。

定义 7.14[155]　（EFD）在关系实例 r 中，当 X 值能够唯一地决定 Y 值时，$X\rightarrow Y$ 是外延的（或具体的）函数依赖 EFD（ExTensional Function Dependency）。

定义 7.15[155]（IFD）　在关系模式 $R(A)$ 中，当 $X\rightarrow Y$ 满足所有的关系实例时，$X\rightarrow Y$ 是强函数依赖（InTensional Function Dependency）。

从定义上看，EFD 是针对单个数据库实例而言的，而强函数以来 IFD 是针对某个数据库模式下所有的数据库实例而言的，本文中，为了便于与信息表比较，把所研究的函数依赖限定为 EFD。

定义 7.16　设 r 为关系模式 R 的实例，X，$Y\subseteq R$ 为其任意两个属性集，r 中 $X\rightarrow$XY 的先行例外集为：$r_{ae}=\{t\in r|\exists t\in r,\ t[X]=t[X]\wedge t[Y]\neq t[Y]\}$。

定义 7.17[156]　设 r 为关系模式 R 的一个实例，X，$Y\subseteq$为其两个属性集。说 $r_e\subset r$ 对 $X\rightarrow Y$ 是一个元组例外关系（或简单地说是例外关系），当且仅当在 r 中：

（1）$(r-r_e)$ 满足 $X\rightarrow Y$。

（2）$\forall t\in r_e$，$(r-r_e)\cup\{t\}$不满足 $X\rightarrow Y$。

（3）$r_e\subset r$ 满足(1)与(2)使得$\#(r'_e)<\#(r_e)$，这里$\#(r_e)$表示关系的元组数。

定义 7.18[63]　设 r 为关系模式的实例 X，$Y\subseteq R$ 为其两个属性集，$r_e\in\varepsilon_{X\rightarrow Y}(r)$，那么 r 满足 α 满足部分依赖 $X\xrightarrow{\alpha}Y$，其中α值为：

$$a=\begin{cases}1 & \text{if} \quad card(r)=0\\ 1-card(rg)/card(r) & \text{otherwise}\end{cases} \qquad (7.1)$$

其中，$\varepsilon_{X\rightarrow Y}(r)$为所以可能的例外关系的集合。

信息系统实际上是数据库关系概念的泛化，不同的是信息系统中不同的对象可以有相同的属性值，而数据库关系则不允许，鉴于此，可以对数据库中的函数依赖及关联规则与信息系统中的决策规则利用粒计算的位表示法进行统一的分析与研究。

1. 关联规则——高频繁模式的软规则

关联规则的提取是要对其支持度和置信度设置相应的阈值的，在关联规则中，“频繁模式”是一个非常重要的概念，因此可以说关联规则是高频繁模式的主要表现形式。设 b 表示信息表的某个属性值，B 表示其对应的颗粒，这样 b=Name(B)，注意 b_i 的频率为 $card(B_i)$，b 的频率为 $card(B)$，这里 $B=\cap_i B_i$。则有下列定理：

引理 7.4[155]

（1）设θ为阈值，如果 $card(B_i)\geqslant\theta$，则 b_i 是一个关联规则；

（2）设θ为阈值，b 是一子元组，如果 $card(B)\geqslant\theta$，则 b_i 是一个关联规则。

命题 7.1 具有最小支持度阈值 S%和最小置信度阈值 C%的关联规则可以利用软规则解释如下：设 $A=\{A_1, A_2, \cdots, A_n\}$，$B=\{B_1, B_2, \cdots, B_m\}$，$A\cap B=\phi$ 为关系数据库的两个任意属性集，当 $card(A\cap B)/card(U)\geqslant S\%$，即 $card(\mathrm{BIT}(A\cap B))/card(\mathrm{BIT}(U))\geqslant S\%$，且 $card(\mathrm{BIT}(A\cap B))/card(\mathrm{BIT}(A))\geqslant C\%$时（$card(\mathrm{BIT}(*))$表示BIT(*)中“1”的个数），则有 $P_A\subseteq Q_B$，关联规则 $A\rightarrow B$ 成立。

对于适当的数据库，利用上述命题可以把对其关联规则的获取转换为利用粒计算位表示来判定和获取。

2. 决策规则——高可信度模式的软规则

如上所述，决策规则可分为确定性规则和不确定性规则两种，而确定性决策规则可以利用粒计算的位方法形式化地描述为：

命题 7.2 设 A，B 为信息表的两个属性，c，d 为 A，B 对应

的两个属性值，而 Neigh(c)，Neigh(d)为 c，d 对应的基本颗粒，当且仅当 $\forall c\in A$，$\exists d\in B$，使得 BIT(Neigh(c))∧BIT(Neigh(d))=BIT(Neigh(c))时，确定性决策规则 $A\rightarrow B$ 成立。

而不确定性规则则是一种高可信模式的软规则，具体地可以形式化地描述如下：

命题 7.3　设 A，B 为信息表的两个属性，c，d 为 A，B 对应的两个属性值，而 Neigh(c)，Neigh(d)为 c，d 对应的基本颗粒，当且仅当$\forall c\in A$，$\exists d\in B$，使得 BIT(Neigh(c))∧BIT(Neigh(d))=BIT(Neigh(c))且 $card$(BIT(Neigh(c)))$\geqslant\theta$ (θ为阈值)时，不确定性决策规则 $A\rightarrow B$ 成立。

3. 外延的函数依赖（EFD）——全可信度模式的软规则

设 A，B 为信息表的两个属性，在经典的关系模型中，通过扫描关系数据库中两列对应的属性值（c_i 与 d_i）来检测函数依赖的存在；在粗糙集理论中，把 A，B 定义为两个等价关系 E_A，E_B，若 $E_A\subseteq E_B$ 则表明函数依赖 $A\rightarrow B$ 成立；在粒计算的位表示法中，则通过判定两个颗粒的位表示的“与”操作的结果来判定函数依赖的存在，具体地可以描述如下：

命题 7.4　设 A，B 为信息表的两个属性，c，d 为 A，B 对应的两个属性值，而 neigh(c)，neigh(d)为 c，d 对应的基本颗粒，当且仅当$\forall c\in A$，$\exists d\in B$，使得 BIT(neigh(c))∧BIT(neigh(d))=BIT(neigh(c)) 且 $card$(BIT(neigh(c)))/$card$(BIT(neigh(c))∧BIT(neigh(d)))=1（1 为阈值）时，$A\rightarrow B$ 成立。

称 EFD 为全可信模式的软规则，是因为它是全依赖性的决策规则，即 K=1。

通过上面的分析，可以得出：

（1）关联规则实际上是一种不确定性决策规则，即根据规定的最小支持度阈值和最小置信度阈值来确定的决策规则。

（2）关联规则和决策规则（尤其是不确定性决策规则）均为

一种受阈值约束的软规则。

（3）外延的函数依赖从根本上讲是一种确定性决策规则，或者说是确定性的软规则。

对关联规则和决策规则的度量公式做一比较，来研究它们之间的关系。关联规则的支持度可表示为 $Support(A\rightarrow B)=P_r(A\cup B)$，可以把它化为：

$$\begin{aligned} Support(A\rightarrow B) &= \frac{Support\_count(A\cap B)}{Support\_count(U)} \\ &= \frac{card(A\cap B)}{card(U)} = 1-\frac{card(\overline{A\cap B})}{card(U)} \end{aligned} \tag{7.2}$$

明显地，$card(A\cap B)+card(\overline{A\cap B})=card(U)$。

其中，*Support_count*(*)表示事务计数。

对于决策规则，按照定义 7.13，有 D 为 K（$0\leqslant K\leqslant 1$）度依赖于 C，记为 $C\Rightarrow_K D$，$K=\gamma(C,D)=\frac{|\mathrm{POS_C}(D)|}{|U|}$，这里 $\mathrm{POS_C}(D)=\mathrm{POS_C}(d_D)$很明显，

$$\begin{aligned} K=\gamma(C,D) &= \sum_{X\in U/D}\frac{|\underline{C}(X)|}{|U|} = \frac{Support\_count(C(x)\cap d_D(x))}{Support\_count(U)} \\ &= \frac{Card(C(x)\cap d_D(x))}{Card(U)} = 1-\frac{Card(\overline{C(x)\cap d_D(x)})}{Card(U)} \end{aligned} \tag{7.3}$$

其中，$Card(\overline{C(x)\cap d_D(x)})$为 $C(x)\cap d_D(x)$的补的数目，显然：

$$Card(\overline{C(x)\cap d_D(x)})+Card(C(x)\cap d_D(x))=Card(U)$$

比较公式（7.1），（7.2）和（7.3）可以得出，关联规则的支持度与决策属性的依赖性度量本质上是相同的，按照粒计算的观点，关联规则、决策规则均为一种颗粒，其支持度、置信度、依赖性度量本质上是一种颗粒的包含关系，因此，公式（7.1）和（7.2）可以利用信息颗粒的包含度量公式统一为：

设α，α'，$\beta\in\{EF_B(x):B\subseteq A\ \&x\in U\}$，$\|\alpha\|_{IS}$，$\|\beta\|_{IS}$ 为 IS（信息系统）中满足α与β的对象的集合[63]，则：

$$
\begin{aligned}
C_{IS}(\alpha\beta)&=\begin{cases}1 & if \quad \alpha=\phi\\ \dfrac{Supporl_{IS}(\alpha,\beta)}{Card(\|\alpha\|_{IS})} & if \quad \alpha\neq\phi\end{cases}\\
&=\begin{cases}1 & if \quad \alpha=\phi\\ \dfrac{Card_{(\|\alpha\|_{IS}\cap,|\beta|_{IS})}}{Card(\|\alpha\|_{IS})} & if \quad \alpha\neq\phi\end{cases}\\
&=\begin{cases}1 & if \quad \alpha=\phi\\ 1-\dfrac{Card(\|\beta\|_{IS})}{Card(\|\alpha\|_{IS})} & if \quad \alpha\neq\phi\end{cases}
\end{aligned}
\tag{7.4}
$$

关联规则的支持度、置信度及α-部分依赖可以利用式（7.4）统一度量。明显地，式（7.1）、（7.2）、（7.4）具有统一的表示形式，因此可以得出一个结论，既关联规则的支持度、置信度及α-部分依赖本质上都是一种包含度量，并且公式（7.4）为它们的统一度量形式[128]。

本节利用粒计算方法对信息系统关联规则、决策规则和函数依赖之间的关系进行了分析，然后利用信息颗粒原理对关联规则度量、决策规则度量和外延的函数依赖度量的关系进行了研究，并建立了这些度量的统一模型。

第 8 章　粗糙函数依赖的近似度量

博学而笃志，切问而近思，仁在其中矣
——孔子《论语》

为了发现粗糙关系数据库中潜在的和有趣的模式，本章提出并研究了粗糙函数依赖的近似度量。

首先，对于关系数据库的近似度量及其满足的性质进行了研究，在此基础上提出了粗糙关系数据库的近似度量及精确度量，对该两种度量进行了形式化定义，并且进一步研究了它所满足的性质，该度量的提出及其性质的研究有利于粗糙关系数据库的知识发现及数据查询的研究，该理论的提出进一步扩大了粗糙关系数据库的研究领域。

8.1　引言

经典的关系数据库理论是由 E.F.Codd 提出的，在他的研究工作中，系统地提出了函数依赖理论和关系规范化理论。在过去的几十年里关系数据库理论得到了长足的发展并应用到数据库知识发现、数据挖掘和模式识别等领域，与相关学科的关系如粗糙集、模糊集、商空间理论、粒计算和关联规则的关系也得到了研究，形成一些新的理论和学科，如粗糙关系数据库、模糊关系数据库、粒计算和商空间等，与函数依赖理论相关的粗糙函数依赖、模糊函数依赖理论也得到不同学者的深入研究。

学者 Chris Giannella 和 Edward Robertson 在深入研究关系数据库理论、函数依赖理论及其度量的基础上，提出了关系数据库

函数依赖的近似度量理论——近似的函数依赖，给出了函数依赖近似度量的一般定义，并且得出了其满足的五个函数依赖近似公理：零公理、对称公理、单调公理、分组公理和“权重和”公理[157]。

美国学者 Theresa Beaubouef 于 20 世纪 90 年代初对关系数据库与粗糙集的关系进行了系统研究，提出粗糙关系数据库[42]、Intuitionistic Rough Sets[91~95]等，并以此为基础对粗糙关系操作算子、粗糙关系数据库的不确定性度量、粗糙函数依赖、粗糙数据查询、模糊关系数据库模型、函数依赖与知识发现等专题进行了研究，并把它们应用于地理信息系统中。在粗糙关系数据库中度量的研究中，Theresa Beaubouef 使用了信息论度量、粗糙熵和粗糙模式熵等[90]。作者近年来在这些问题上也进行了一些研究[106~123]。由于粗糙关系数据库是多值的，其信息具有不确定性和含糊性，所以具有某些不具备函数依赖的数据间有可能存在着一种近似依赖，在 Theresa Beaubouef 及其他学者的研究中，并没有关于粗糙函数依赖近似度量的研究，本章中将给出粗糙函数依赖近似度量及精确度量的形式化研究，并对其满足的基本性质进行初步研究，本章的研究将进一步扩大粗糙关系数据库的研究领域，并促进粗糙关系数据库的知识发现的研究。

8.2　相关工作

函数依赖是数据库关系中属性之间的关系体现：函数依赖表明关系中一个属性值可以由另外其他的属性值唯一决定。

一般来说，函数依赖是关系模式 R 上的表达式 $X\rightarrow A$，其中，$X\subseteq R$ 且 $A\in R$。函数依赖对于 R 中一个给定的关系实例 r 成立，如果对 r 中所有的元组 t，$u\in r$ 则有：若 $t[B]=u[B]$，$B\in X$，那么 $t[A]=u[A]$（也说 t 与 u 在 X 与 A 出相同）。

在文献[157]中，作者研究了近似的函数依赖如何度量的问题。其主要动机在于一个数据库表中近似的函数依赖可以表示潜

在的有兴趣的模式。发现的这种有趣的模式是一个很有价值的数据挖掘问题。作者首先提出了一个近似度量公式用于表示下列问题：对于关系数据库表 T 中近似函数依赖 $X\rightarrow Y$ 的度量可以用 $\prod_X(T)$对$\prod_Y(T)$的函数来表示这种度量。在此基础上，对于所提出的度量是否满足一组公理进行了证明，并且把所提出的度量与其它文献中的两种度量进行了比较，最后把其所提出的度量用几种真实的数据集进行了测试，表明所提出的度量相比其他的度量更加实用有效。

在文献[157]中，作者给出了函数依赖近似度量的一般定义。用更一般的术语表示，T 中函数依赖 $X\rightarrow Y$ 的近似度量应该是对每个 $T_{X=x}$，$x\in\Pi_X(T)$的权重和。给定 $y\in\Pi_y(T)$，设 $f_{Y\backslash X}(y\backslash x)$为 y 对于 x 的相关频率：$f_{Y\backslash X}(y|x)=C_{XY}(x,\ y)/C_X(x)$，这里 $C_X(x)$为 X 中 x 的计数 X: $T_{X=x}$中元组的个数。对 $y\in\Pi_Y(T)$，设 $C_{XY}(x,\ y)$为 T 中元组 t 的数目，其中，$t[X\cup Y]=(x,\ y)$。与 x 的相关联的频率矢量为 $[f_{Y\backslash X}(y\backslash x):y\in\Pi_Y(T_{X=x})$。进一步，$x$ 值的相关频率矢量记为 $[f_X(x):x\in\Pi_X(T)]$。

作者进一步提出关系数据库函数依赖近似度量有以下 5 个公理：

（1）**零公理** $\Gamma_1([1])=0$；

（2）**对称公理** 对所有的 $1\leqslant q$，$1\leqslant i\leqslant j\leqslant q$，$\Gamma q([\cdots,\ f_i,\ \cdots,\ f_j,\ \cdots])=\Gamma q([\cdots,\ f_j,\ \cdots,\ f_i,\ \cdots])$；

（3）**单调公理** 对所有的 $q'\geqslant q\geqslant 1$，$\Gamma_{q'}(1/q',\ \cdots,\ 1/q')\geqslant\Gamma_q([1/q,\ \cdots,\ 1/q])$；

（4）**分组公理** 对所有的 $q\geqslant 3$，$\Gamma_q([f_1,\ \cdots,\ f_q])=\Gamma_{q-1}([f_1,\ \cdots,\ f_{q-2},\ f_{q-1}+f_q])+(f_{q-1}+f_q)\Gamma_2([f_{q-1}/f_{q-1}+f_q],\ f_q/f_{q-1}+f_q)]$；

（5）**权重和公理** 对所有的 $p\geqslant 2$ *and* $q_1,\ \cdots,\ q_p\geqslant 1$，$\Gamma_{p,q_1,\cdots,q_p}([f_1,\ \cdots,\ f_q],\ [f_{1|1},\ \cdots,\ f_{q1|1}],\ \cdots,\ [f_{1|p},\ \cdots,\ f_{qp|p}]=\sum_{i=1}^{p}f_i\ \Gamma_{1,q_i}([f_{1|i},\ \cdots,\ f_{q_i|i}])$。

Theresa Beaubouef 在提出粗糙关系数据库后，又对粗糙关系数据库的度量进行了研究，在文献[90]和[105]中，作者给出了粗糙关系数据库粗糙熵、粗糙模式熵及粗糙关系熵的一般定义。

定义 8.1[90]　一个粗糙集合 X 的粗糙熵 $E_r(X)$为：

$E_r(X) = -(\rho_R(X))[\sum Q_i \ \log(P_i)]$　i=1，…，n 为等价类，$\rho_R(X)$为集合 X 的粗糙度。C_i 为等价类 X 的的基数或包含在该等价类中的元素数目等价类 i 及所有给定的等价类的分布概率是相等的，P_i=1/C_i 表示等价类中值的分布概率，Q_i 表示等价类 i 在论域中的分布概率。

定义 8.2[105]　（粗糙模式熵）设 R={A_1, A_2, …, A_m}是一个关系模式，其中，$dom(A_i)$有 n_i 个等价类：$X_i^{(1)}$，…，$X_i^{(n_i)}$，i=1，…，m，则 R 的粗糙模式熵为：

$$E_S(R)=-\sum_{i=1}^{m}(\sum_{j=1}^{i}(|X_i^{(j)}|/|dom(A_i)|)\log(1/|X_i^{(j)}|))$$

其中，$|X_i^{(j)}|$及$|dom(A_i)|$分别表示 $X_i^{(j)}$ 及 $dom(A_i)$中的元素个数。

定义 8.3[105]　设 R={A_1，A_2，…，A_m}是一个关系模式，其中，属性 A_i 的域 $dom(A_i)$的等价类为 $X_i^{(1)}$，…，$X_i^{(n_i)}$，r 是 R 上的一个粗糙关系，用$|r|$表示 r 中的元组个数，用$|t(A_i)|$表示元组 t 在属性 A_j 上取值集合的元素个数，令 r_i={ $t(A_i)|t\in r$}，$r_i^{(j)}=r_i\cap X_i^{(j)}$，则

$$\|r_i\|=\sum_{t\in r}|t(A_i)|$$

以及

$$\|r_i^{(j)}\|=\sum_{t\in r}|t(A_i)\cap X_i^{(j)}|$$

则 r 的粗糙关系熵定义为：

$$E_R(r)=-\sum_{i=1}^{m}D_{\rho_i}(r)(\sum_{j=1}^{n_i}DQ_i^j\log DP_i^j)$$

其中，

$$D_{\rho_i}(r)=1-\frac{\| r_i^0 \|}{\| r_i \|}$$

这里，$r_i^0=\{t|t\in r, |t(A_i)|=1\}$

$$DQ_i^j=\| r_i^{(j)} \| / \| r \|$$

$$DP_i^j=\| r_i^{(j)} \| / \| r_i \|$$

在该定义中，认为具有以下形式的上近似及下近似：若 S 是一个粗糙关系，则 S 的上近似就是 S 本身，S 的下近似是 S 中所有在个属性上只取单个值的那些元组组成的集合，于是定义中的 $\| r_i^{(0)} \| / \| r_i \|$ 可以理解为精确度，$D_{\rho_i}(r)$ 可以理解为粗糙度。另外，定义中的 DQ_i^j 可以理解为属性 A_i 上关系 r 取 $X_i^{(j)}$ 等价类中的值的概率，DP_i^j 可以理解为属性 A_i 上关系 r 取 $X_i^{(j)}$ 等价类中的值的相对概率。

8.3 粗糙函数依赖（RFD）的度量

1. RFD的近似度量

文献[42]给出了粗糙函数依赖的定义：

定义 8.4 设 X，Y 为粗糙关系模式 R 的属性子集，粗糙函数依赖（RFD）$X\rightarrow Y$ 对于一个粗糙关系模式 R 的所有实例 T 都成立，当满足：

（1）对任意两个元组 t，$t'\in \underline{R}\ T$，Redundant($t(X)$，$t'(X)$)→Redundant($t(Y)$，$t'(Y)$)且

（2）对任意两个元组 s，$s'\in \overline{R}\ T$，Rough-redundant($s(X)$，$s'(X)$)→Rough-redundant($s(Y)$，$s'(Y)$)。

对于表 8-1，若按照经典的函数依赖及粗糙函数依赖的定义，属性 COUNTRY 与 FEATURE 之间是没有依赖关系的，但是若仔

细观察，它们之间是有一种近似依赖关系的。比如 x_1，x_2，x_3 中，属性“COUNTRY”相应的属性值为“US”，“FEATURE”属性中有子属性值“MARSH”与之对应，对于 X_6，其元组是唯一的；对 x_7，x_8，x_9，属性“COUNTRY”对应的属性值为“MEXICO”，而其对应的“FEATURE”属性中有子属性值“SAND”与之对应，x_{10}，x_{11} 所对应的元组均是唯一的；只有 x_4，x_5 不满足粗糙函数依赖。综上所述，可以看出，在属性“COUNTRY”与“FEATURE”之间是有一种近似的依赖关系存在的。

表 8-1　地理信息

OBJ	ID	COUNTRY	FEATURE
x_1	U123	US	{MARSH，LAKE}
x_2	U124	US	MARSH
x_3	U125	US	{MARSH，PASTURE，RIVER}
x_4	U126	US	{FOREST，RIVER}
x_5	U147	US	{SAND，ROAD，URBAN}
x_6	U157	{US，MEXICO}	{SAND，ROAD}
x_7	M007	MEXICO	{SAND，ROAD}
x_8	M008	MEXICO	SAND
x_9	M009	MEXICO	SAND
x_{10}	CO39	BELIZE	JUNGLE
x_{11}	CO40	{BELIZE，INT}	{JUNGLE，BEACH，SEA}

定义 8.5　给定任意整数 p，q_1，…，$q_p \geqslant 1$，设 $\Pi_C(T)=\{c_1, c_2, \cdots, c_p\}$，这里 Π 为数据库的投影操作，c_i 为属性 C 对应的属性值。$|\Pi_F(T_{C=c_i})|=q_i$ 为属性值为 c_i 时属性 F 所对应的投影数目，f_i 为频率矢量且 $f_{j|i}$ 表示与每个 c_i 值关联的 F 的相关频率矢量。$\sum_{j=1}^{q_i} f_{j|i} = f_i$ 且 $\sum_{j=1}^{p} f_j = 1$，粗糙函数依赖的近似度量可以

定义为：

$\Gamma_{p,q_1,\cdots,q_p}(f_1, f_2, \cdots, f_{q_p})=\sum_{i=1}^{p} f_i(1-f_{j|i})$。

表 8-1 中近似函数依赖 COUNTRY→FEATURE 的程度为：$\Gamma_{p,q_1,\cdots,q_p}(f_1, f_2, \cdots, f_{q_p})=\sum_{i=1}^{p} f_i(1-f_{j|i})=\Gamma_{5,5,1,3,1,1}(5/11, 1/11, 3/11, 1/11, 1/11)=5/11*(1-3/5)+1/11*(1-1)+3/11*(1-1)+ 1/11*(1-1)+1/11*(1-1)=2/11$。

2. 近似度量性质

在相关工作中，介绍了关系数据库的五个基本性质，下面研究粗糙函数依赖近似度量所满足的基本性质。

（1）**零公理** 粗糙函数依赖的近似度量满足零公理，即当粗糙函数依赖成立时，其近似度量值为零。

证明：当粗糙函数依赖 $X\rightarrow Y$ 成立时，按照定义 8.5，

$\Gamma_{p,q_1,\cdots,q_p}(f_1, f_2, \cdots, f_{q_p})=\sum_{i=1}^{p} f_i(1-f_{j|i})=\sum_{i=1}^{p} f_i(1-1=0)$，

即粗糙函数依赖的近似度量值为 0，因此，该度量满足零公理

（2）**对称公理** 可以形式化地表述为：对所有的 $q\geqslant 1$ 且 $1\leqslant g\leqslant k\leqslant q$，有 $\Gamma_q([\cdots, f_g, \cdots, f_k, \cdots])=\Gamma_q([\cdots, f_k, \cdots, f_g\cdots])$。

它意味着频率矢量在近似度量公式中出现的顺序将不影响最终的度量值。

证明：按照定义 8.5，因为 $\Gamma_{p,q_1,\cdots,q_p}(f_1, f_2, \cdots, f_g, \cdots, f_k, \cdots, f_{q_p})=\sum_{i=1}^{p} f_i(1-f_{j|i})=f_1(1-f_{j|1})+f_2(1-f_{j|2})+\cdots+f_g(1-f_{j|g})+\cdots+f_k(1-f_{j|k})+\cdots+f_p(1-f_{j|p})=f_1(1-f_{j|1})+ f_2(1-f_{j|2})+\cdots+f_k(1-f_{j|k})+\cdots+f_g(1-f_{j|g})+\cdots+f_p(1-f_{j|p})=\Gamma_{p,q_1,\cdots,q_p}(f_1, f_2, \cdots, f_k, \cdots, f_g, \cdots, f_{q_p})$。因此对称公理成立。

（3）**单调公理** 假设当函数依赖成立时其近似度量将返回值 0，那么在表 8-2 中的近似依赖度量值不应该大于表 8-3 中的近似依

赖度量值形式化地描述为：对所有的 $q'\geqslant q\geqslant 2$，$\Gamma_{q'}([1/q',\cdots,1/q'])\geqslant\Gamma_q([1/q,\cdots,1/q])$。粗糙函数依赖的近似度量（定义 8.5）满足单调公理。如对表 8-2，$\Gamma_q([1/q,\cdots,1/q])=\Gamma_2(1/2,1/2)=1*(1-1/2)=1/2$，对表 8-3，$\Gamma_{q'}([1/q',\cdots,1/q'])=\Gamma_3(1/3,1/3,1/3)=1*(1-1/3)=2/3$，明显地，$\Gamma_{q'}([1/q',\cdots,1/q'])=2/3\geqslant\Gamma_q([1/q,\cdots,1/q])=1/2$。

表 8-2　粗糙关系 1

A	B	C
1	(1，1)	1
1	(1，1)	2
1	(2，2)	3
1	(2，2)	4

表 8-3　粗糙关系 2

A	B	C
1	(1，1)	1
1	(1，1)	2
1	(1，1)	3
1	(2，2)	4
1	(2，2)	5
1	(2，2)	6
1	(3，3)	7
1	(3，3)	8
1	(3，3)	9

（4）**权重和公理**　对于所有的 $p\geqslant 2$ 及 $q_1,\cdots,q_p\geqslant 1$，$\Gamma_{p,q_1,\cdots,q_p}([f_1,\cdots,f_q],[f_{1|1},\cdots,f_{q1|1}],\cdots,[f_{1|p},\cdots,f_{qp|p}]=\sum_{i=1}^{p}f_i\,\Gamma_{1,q_i}([f_{1|i},\cdots,f_{q_i|i}])$。比较“定义 8.5”与关系数据库的

权重和公理，它们的实质是一样的，因此函数依赖的近似度量满足权重和公理。

3. RFD的精确度量

设α与Γ分别为粗糙函数依赖的精确度量算子与近似度量算子，那么

$\alpha=1-\Gamma_{p,q_1,\cdots,q_p}(f_1，f_2，\cdots，f_{q_p})=1-\sum_{i=1}^{p} f_i(1-f_{j|i})$。

在表 8-1 中，粗糙函数依赖 COUNTRY→FEATURE 的精度度量可以用下式计算：

$\alpha=1-\Gamma_{p,q_1,\cdots,q_p}(f_1，f_2，\cdots，f_{q_p})= 1-\Gamma_{5,5,1,3,1,1}(5/11，1/11，3/11，1/11，1/11) =1-2/11=9/11$。

有了上述对粗糙函数依赖度量的形式化定义，可以计算出粗糙关系数据库中任意两个属性之间的依赖关系，或研究它们之间是否存在度量关系，这对于研究粗糙关系数据库的知识发现是十分重要的。

命题 8.1 设 α 与 Γ 分别为粗糙函数依赖 $X\rightarrow Y$ 的精确度量算子与近似度量算子，那么

$$\alpha+\Gamma=1$$

按照前面的定义。该结论是明显的。

推论 8.1 设 α 与 Γ 分别为粗糙函数依赖 $X\rightarrow Y$ 的精确度量算子与近似度量算子，如果Γ=0(即α=1)，那么粗糙函数依赖 $X\rightarrow Y$ 成立。

该推论可以从零公理及命题 1 得出。

命题 8.2 度量公式 8.5 对于关系数据库同样成立。

命题 8.3 设 α 与 Γ 分别为粗糙函数依赖 $X\rightarrow Y$ 的精确度量算子与近似度量算子，那么 $0\leqslant\alpha\leqslant1$，$0\leqslant\Gamma\leqslant1$。

8.4　本章小结

本章研究了粗糙关系数据库的近似度量及精确度量，对该两种度量进行了形式化定义，并且研究了它所满足的性质，该度量的提出及其性质的研究有利于粗糙关系数据库的知识发现及数据查询的研究，该理论的提出进一步扩大了粗糙关系数据库的研究领域。进一步，将研究与度量相关的算法以及将本文所提出的度量公式应用相应的数据查询和知识发现中，并且可以开展本文算法与其他对于粗糙关系数据库进行度量的公式的比较研究。

第9章　结　　语

路漫漫其修远兮，吾将上下而求索

——屈原《离骚》

粗糙集理论是20世纪80年代初由波兰数学家Zdzislaw·Pawlak提出的一个分析数据的数学理论，粒计算是近年来新兴的一个软计算方法。由于对于粗糙关系数据库及相关的理论问题研究时间不长，所以尚有许多理论与应用问题等待解决，本书以粗糙集理论和粒计算理论为基础，对粗糙关系数据库和关系数据库中的一些基础理论和度量问题进行了深入的研究，获得了一些初步的研究成果。

9.1　主要结论

（1）系统地研究了关系数据库理论和粗糙集理论之间的关系，在研究与完善粗糙关系数据库（RRDB）的粗糙关系操作算子基础上，提出了粗糙分解算子，并讨论了它的性质。首先以粗糙数据分析技术为工具，对关系数据库理论和粗糙集理论之间的关系进行了系统地研究，然后以粗糙关系数据库模型为基础结合Pawlak代数对粗糙关系操作算子进行了分析，提出了粗糙分解算子的概念，研究了其性质，并对粗糙关系数据库模型与关系数据库模型进行了系统的比较，同时对RRDB与FRDB关系的进行了系统研究。

（2）粗糙关系数据库的数据查询方面，在研究粗糙关系数据库模型的基础上提出了相应的分解原理及查询原理。以粗糙关系

数据库模型为背景，从分解原理、投影原理和粗糙关系数据库的可定义性等几方面讨论了 RRDB 的查询原理，并以此为基础研究了 RRDB 的数据查询，把其数据查询分为精确查询、粗糙（Rough）完全查询和粗糙（Rough）组合查询三类，并从这三方面对粗糙数据查询进行了讨论与仿真实验，仿真结果验证了查询思想的可行性和正确性；同时对 RRDB 与 NIS 的关系和 RRDB 属性值的粗集表示进行了探讨。

（3）以粗糙集理论和关系数据库理论为基础，从函数依赖、范式理论和 Armstrong 公理等方面系统地研究了粗糙关系数据库（Rough Relational Database，RRDB）与模糊关系数据库（Fuzzy Relational Database，FRDB）之间的关系。结果表明，模糊函数依赖与粗糙函数依赖均为经典函数依赖的泛化，模糊范式理论为经典范式的扩充，而粗糙范式理论自成体系，从推理规则上看，它们都不同程度地符合 Armstrong 公理。

（4）分析了 Shoji Hirano 的基于粗糙集的聚类方法的不足，在此基础上提出了利用模糊隶属度、信息颗粒与粗糙集理论相结合进行数据库混合数据聚类的改进方法。首先讨论了信息颗粒和信息粒度的基本原理，提出了字符颗粒和数字颗粒的概念，研究了采用信息颗粒和粗糙集理论进行聚类的机理，并且给出了对纯字符数据和混合数据进行聚类的算法，每种算法又分自然聚类和按照要求的聚类数聚类。

（5）信息系统函数依赖方面，在研究面向机器的数据挖掘模型的基础上提出一种利用信息颗粒位表示进行信息系统函数依赖判定的方法。首先探讨了 T.Y.Lin 提出的信息颗粒位表示及面向机器的数据挖掘模型，研究了关于位表示的性质，然后利用信息颗粒的位表示方法研究了信息系统的函数依赖、恒等依赖、部分依赖的信息颗粒原理，得出了它们相关的性质。该方法可以快速判定和度量信息系统的函数依赖关系，对于信息颗粒采用其位表示，使得数据格式更接近机器的内部表示，且能够直接挖掘信息

系统所蕴含的各种模式。

（6）提出了一种利用粗糙集原理判定关系数据库范式的新方法：首先给出了 1NF（First Normal Form），2NF（Second Normal Form），3NF（Third Normal Form），BCNF（Boyce-Codd Normal Form）的基本概念及粗糙集理论的基本原理，然后利用粗糙集理论研究了给定关系满足某一范式时具有的性质以及判定任意关系所属的范式级别的方法，本文的方法是粗糙集原理应用的一个新的拓展。利用信息颗粒的位表示（Bit Representations）来进行信息系统软规则及其度量之间关系的研究。具体地，首先利用软规则对关联规则、决策规则、函数依赖之间的关系进行了分析，然后对关联规则度量、决策规则度量、外延的函数依赖度量的关系进行了研究，并且建立了这些度量的统一模型。

（7）为了发现粗糙关系数据库中潜在的、有趣的模式提出了粗糙函数依赖的精确度量与近似度量。经典的关系数据库的近似度量被简单回顾，在此基础上提出了粗糙关系数据库的精确度量与近似度量，进一步给出了其形式化定义，研究了度量满足的性质，粗糙函数依赖的近似度量理论的引入将促进粗糙关系数据库知识发现的研究，扩大其研究领域。

9.2 研究展望

本书中提出的一些理论还有待于研究深入探讨，甚至有些缺陷，在粗糙关系数据库的理论和应用方面还有许多问题值得探讨，这些都是以后需要努力研究的地方，近期研究工作包括以下几个方面：

（1）粗糙关系数据库的粗糙查询与分解问题。本书提出的粗糙分解算子还有哪些性质？如何对粗糙关系数据库进行粗糙分解？研究高效的查询与分解算法以利于对粗糙关系数据库的各种查询是值得深入探讨的问题。

（2）对信息系统软规则及其度量之间关系的研究，但是软规则具体如何分类、它还包含哪些规则；如何利用关联规则、决策规则、函数依赖之间的统一模型进行知识发现均是值得深入研究的问题。

（3）只是从理论方面研究了利用粗糙集原理判定关系数据库范式的方法，但是并没有给出具体的利用粗糙方法与技术来判定一个具体的数据库关系范式的算法，与其他理论判定数据库关系范式的比较也没有进行。

（4）对于粗糙关系数据库的度量问题的研究。到目前为止，对于粗糙关系数据库的度量研究方面的文献还比较少，其度量问题也是一个比较复杂的问题，需要进行更深一步的研究。

（5）对于粗糙关系数据库知识发现的研究，这也是一个值得深入研究的专题。

主要符号表

符号	含义
$\underline{apr}_{\mathrm{R}}(X)$	X 由等价关系 R 支持的下近似算子
$\overline{apr}_{\mathrm{R}}(X)$	X 由等价关系 R 支持的上近似算子
$card(X)$ 或 $\|X\|$	集合 X 的势，或称为基数
IND(A)	由属性集 A 所诱导的等价关系，或不可区关系
IND(a)	由属性 a 所诱导的等价关系，或称为不可区分关系
$P\rightarrow Q$ 或 $P\Rightarrow Q$	属性集 Q 依赖于属性集 P
$S=(U, \mathrm{AT}, V, f)$或 $I=$（U，A）	信息系统
$S=(U, \mathrm{AT}, d)$	决策信息系统或称为决策表
SIM(A)	由属性子集 A 所决定的相似关系
$\|\|t\|\|$	满足性质 t 的所有对象的全体
(U, IND(A))	由等价关系 IND(A)诱导的近似空间
$U\times Q$	由集合 U 和 Q 所决定的笛卡尔积
$[x]_A$	由等价关系 IND(A)所决定的对象 x 所在的等价类
α_R(X)	集合 X 在关系 R 下的近似精度
ρ_R(X)	集合 X 在关系 R 下的粗糙度
$\mu_X^A(\cdot)$	集合 X 关于属性集 A 的隶属函数

参 考 文 献

[1] 史忠植著. 知识发现. 清华大学出版社，2002 年 1 月，143～145

[2] 张文修，梁怡著. 不确定性推理原理. 西安交通大学出版社，1996.10

[3] Z.Pawlak, Rough sets, International Journal of Information and Computer Science, 1982,11 (5):341～356

[4] Z. Pawlak, Rough set: Theoretical Aspects of Reasoning About Data, Dordrecht: Kluwer Academic Publishers,1991

[5] 张文修，吴伟志. 粗糙集理论介绍和研究综述. 模糊系统与数学，2000，Vol.14，No.4

[6] 梁吉业. 关于粗糙集度量与粗糙计算方法的研究. 西安交通大学博士学位论文，2001 年，1～10

[7] 张文修，姚一豫，梁怡主编. 粗糙集与概念格. 西安交通大学出版社，2006.7

[8] 李德玉. 粗糙集的代数结构与信息系统的知识约简. 西安交通大学博士学位论文，2002 年，1～20

[9] 王军. 数据库知识发现的研究. 中科院计算所. 1997 年，31～33

[10] Ziarko W.,Variable precision rough set model, Journal of Computer and System Sciences, 1993, 46:39～59

[11] Malcolm Beynon,Reducts within the variable precision rough sets model:A further investigation, European Journal of Operational Research, 2001, 134:592～605

[12] Dubois D,Prade H., Rough fuzzy sets and fuzzy rough sets, International Journal of General Systems,1990,17:191～209

[13] Banerjeem M. and Pal S.K., Roughness of a rough set, Information Sciences, 1996, 93(3-4):235～246

[14] Dubois D,Prade H.,Twofold fuzzy sets and rough sets-some issues in knowledge representation, Fuzzy sets and System,1987,23:3～18

[15] Chakrabarty K.,Biswas R. and Nanda S.,Fuzziness in rough sets,Fuzzy Sets and System,2000,110(1,2):247～251

[16] Beaubouef T. and Perty F.E.,Fuzzy rough set techniques for uncertainty processing in a relational database,International Journal of Intelligence Systems,2000,15(5):389～424

[17] Slowinski R. and Vanderpooten D.,A generalized definition of rough approximations based on similarity,IEEE Transactions on Knowledge and Data Engineering, 100, 12(2):331～336

[18] Daijin Kim,Data classification based on tolerant rough set,Pattern Recognition, 2001, 34:1613～1624

[19] Y.Y.Yao.and Lin T.Y.,generalization of rough sets using modal logic, Intelligence Automation and Soft computing,1996,2:103～120

[20] Y. Y. Yao.,Relational interpretations of neighborhood operators and rough sets approximation, Information Science,111(1-4):239～259

[21] Quafafou M.,α-RST:a generalization of rough set theory, Information Science,2000, 124(1-4):301～316

[22] Grec S.,Matarzzo B. and Slowinski R.,Rough approximation of a preference relation by dominance relations,European Journal of OperationalResearch,1999,117(1):63～83

[23] Kryszkiewicz M., Rough set approach to incomplete information systems, Information Science,1998,112 (1-4):39～49

[24] Kryszkiewicz M., Rule in incomplete information systems, Information Science, 1999,113(3-4):271～292

[25] Chmielewski M.R. and Grzymala-Busse W., Global discretization of continuous attributes as preprocessing for machine learning,International Journal of Approximate Reasoning, 1996,15(4):319～331

[26] 王国胤. Rough 集理论与知识获取. 西安：西安交通大学出版社，2001

[27] Yao Y.Y., Lingras P, Interpretations of belief functions in the theory of rough set, Information Science,1998,104:81～106

[28] Bonikowski Z., Algebraic structures of rough sets. In:Ziarko W. (Ed.), Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-Verlag, 1994, 242～247

[29] Bryniarski E., Formal description of rough sets, In: Ziarko W. (Ed.), Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-Verlag, 1994, 208～216

[30] Polkowski L.,On convergence of rough sets, In:Slowinski R. (Ed.), Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory, Dordrecht: Kluwer Academic, 1992, 305～311

[31] Li D. Y. and Ma Y. C., Invariant characters of information systems under some homomorphisms, Information Sciences,2000, 129(1-4): 211～220

[32] Kuroki N., Rough ideals in semigroups, Information Sciences,1997, 100:139～163

[33] Wong S. K. and Ziarko W., On optimal decision rules in decision tables, Bulletin of Polish Academy of Sciences, 1985,33: 693～696

[34] Shan N. and Ziarko W., An incremental learning algorithm for constructing decision rules, In: Ziarko W. (Ed.),Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-verlag, 1994,326～334

[35] Shan N. and Ziarko W., Data-based acquisition and incremental modification of classification rules, Computational Intelligence,1995, 11(2): 357～369

[36] Hu X. and Cercone N., Learning in relational database: a rough set approach, Computational Intelligence, 1995, 11(2): 323～338

[37] Jelonek J. et.al., Rough set reduction of attributes and their domains for neural networks, Computational Intelligence, 1995, 11(2): 339～347

[38] Muraszkieqicz M. and Rybinski H., Towards a parallel rough sets computer,In: Ziarko W. (Ed.), Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-Verlag, 1994, 167～181

[39] Duntsh I., A logic for rough sets,Theoretical Computer Science, 1997,

179(1,2): 427～436

[40] T.Y.Lin.An Overview of Rough Set Theory from the point of View of Relational Databases, Bulletin of Internal Rough Set Society Volume 1,Number 1:30～34

[41] J.W.Guan,D.A.BELL.Rough Computational methods for information systems, Artificial Intelligence [J].105(1998),77～103

[42] Beaubouef Theresa,Frederick E.Petry, Bill P. Buckles.Extension of the relational database and its algebra with rough set techniques.Computational Intelligence, 1995, 11(2):233～245

[43] Theresa Beaubouef.Uncertainty processing in a relational database model via a rough set representation, University Microfilms International.A Bell&Howell Information Company.Doctor dissertation,67～76,1994

[44] Golan R. and Ziarko W.,Methodology for stock market analysis utilizing rough set theory,Proceedings of IEEE/IAFE Conference on Computational Intelligence for financial Engineering,New Jersey,1995,32～40

[45] Z.Pawlak.et.al., Rough Sets, Communications of the ACM,1995, 38(11):89～95

[46] Tsumoto S., Automated extraction of medical expert system rules from clinical databases based on rough set theory, Information Sciences, 1998, 112(1-4): 67～84

[47] Teghem J.et.al.,Use of rough sets method to draw premonitory factors for earthquakes by emphasizing gas geochemistry, In:Slowinski R.(Ed.), Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory, Dordrecht:Kluwer Academic Publishers, 1992,165～179

[48] Nejman D.,A rough set based method of handwritten numerals classification, Institute of Computer Science Reports,Warsaw University of Technology, Warsaw, 1994

[49] Slowiski R., Rough set approach to decision analysis. AI Expert, 1995,

March: 19～25

[50] Z.Pawlak, Rough set approach to knowledge-based decision support. European Journal of Operational Research, 1997, 99(1): 48～57

[51] Slowiski R., Zopounidis C. and Dimitras A. I.,Prediction of company acquisition in Greece by means of the rough set approach. European Journal of Operational Research,1997, 100(1): 1～15

[52] Peng C.,Multi-valued neural network and knowledge acquisition method by the rough sets for ambiguous recognition problem. Pceedings of IEEE International Conference on System,Man and Cybenetics, Beijing, 1996, 736～740

[53] Yasdi R., Combining rough sets learning and neural learning-method to deal with uncertain and imprecise information. Neurcomputing, 1995, 7(1):61～84

[54] Plonka L. and Mrozek A., Rule-based stabilization of the inverted pendulum. Computational Intelligence,1995,11(2):348～356

[55] Czogala E.et.al., Idea of a rough fuzzy controller and its application of the stabilization of a pendulum-car system. Fuzzy Sets and Systems,1995,72(1)

[56] Mrozek A., Rough sets and dependency analysis among attributes in computer implementations of experts’s inference models. International Journal of Man-Machine Studies, 1989,30(4): 457～473

[57] Z.Pawlak.,An inquiry into anatomy of conflicts, Information Science,1998, 109(1-4):65～78

[58] 王珏. Rough Set 理论对归纳机器学习的贡献. 计算机科学，2001，28（5,专刊）

[59] Y.Y.Yao, Constructive and algebraic methods of the theory of rough sets, Journal of Information Sciences, 1998, 21～47

[60] Z.Pawlak. and Skowron A., Rough membership function,In:Yaeger R. R. et.al. (Eds.), Advances in the Dempster Shafer Theory of Evidence, John Wiley and Sons, Singapore, 1994, 251～271

[61] Lotfi A.Zadeh, Fuzzy Sets and Information Granularity, In:M.Gupta, R.Ragade, R. Yager,(Eds.), Advances in Fuzzy set theory and Applications, North-holland, Amsterdam,1979,3～18

[62] Skowron A. and Stepaniuk J., Constructive information granules. In: Proc. of the 15th IMACS World Congress on Scientific Computationm, Modelling and Applied Mathematics. Berlцín, Germany, 24～29 August 1997, Artificial Intelligence and Computer Science 4, 1997, 625～630

[63] Skowron A. and Stepaniuk J.,Information granules and approximation spaces,In: Proc. 7th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU98), Paris, La Sorbonne, 6-10 July 1998, 1354～1361

[64] Skowron A. and Stepaniuk J.,Peters J.F.: Approximations of Information Granule sets,(Eds.) Zizrko W.,Yao Y.Y.,Rough Sets and Current Trends in computing (RSCTC’ 2000), Banff, Alberta, Canda, 2000, 33～39

[65] Andrzej Skowron,Jaroslaw Stepaniuk., Information Granules:Towards Foundations of Granular Computing,International Journal of Intelligent Systems, Vol 16, 2001, 57～85

[66] Polkowski L. and Skowron A., Rough mereology,Proceedings of the Symposium on Methodologies for Intelligent Systems, Charlotte, NC, Octomber 16-19, Lecture Notes in AI, Vol.869, Springer-Verlag, Berlin, 1994, 295～301

[67] Lotfi A.Zadeh, Fuzzy graphs, rough sets and information granularity, Third International Workshop on Rough Sets and Soft Computing, November 10～12, 1994, San, Jose, CA(invited lecture)

[68] Lotfi A.Zadeh, information granulation, fuzzy logic and rough sets, Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, November 6～8, 1996, Tokyo, Japan (invited lecture)

[69] Lotfi A.Zadeh, Fuzzy logic = computing with words, IEEE Trans. On Fuzzy Systems , vol. 4, no. 2, 1996:103～111

[70] Lotfi A.Zadeh, Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic, Fuzzy Sets and Systems, vol. 90, no. 2, 1997, 111～128

[71] T.Y.Lin., Theoretical Sampling for Data Mining, In: Proceeding of 14th Annual International Symposium Aerospace/Defense Sensing, Simulation, and Controls , SPIE Vol 4057, Orlando, 2000, April 24～28, 192～200

[72] Eric Louie and T.Y.Lin,A Data Mining Approach using Machine Oriented Modeling: Finding Association Rules using Canonical Names, In: Proceeding of 14th Annual International Symposium Aerospace/Defense Sensing, Simulation, and Controls , SPIE Vol 4057, Orlando,2000, April 24～28, 148～154

[73] T.Y.Lin, Granular Computing: Fuzzy Logic and Rough Sets, In: Computing with words in information/intelligent systems, L.A. Zadeh and J. Kacprzyk (eds), Springer- Verlag , 1999, 183～200

[74] T.Y.Lin, Granular Fuzzy Sets:A View from Rough Set and Probability Theories, International Journal of Fuzzy Systems, Vol.3, No.2, 2001, June, 373～381

[75] T.Y.Lin.,Granular Computing on Binary Relations I: Data Mining and Neighborhood Systems, In:Rough Sets in Knowledge Discovery, A. Skoworn and L. Polkowski (eds), Springer-Verlag, 1998, 107～121

[76] T.Y.Lin, Association Rules in Semantically Rich Relations: Granular Computing Approach , Proceedings of International Workshop on Rough Set Theory and Granular Computing, Bulletin of International Rough Set Society, Matsue, Shimane, Japan2001 May 20～22, 143～149

[77] T.Y.Lin,Data Modeling for Data Mining,Data Mining and Knowledge Discovery: Theory, Tools and Technology IV,Belur V.Dasarathy,Editor, Proceedings of SPIE Vol. 4730, 2002, 138～145

[78] Y.Y.Yao, Stratified Rough Sets and Granular Computing, Proceedings of the 18th International Conference of the North American Fuzzy

Information Processing Society, New York, USA, June 10-12, 1999, Dave, R.N. and Sudkamp. T. (Eds.), IEEE Press, 800～804.

[79] Y.Y.Yao, Rough sets, neighborhood systems, and granular computing, Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering, Edmonton, Canada, May 9-12, 1999, Meng, M. (Ed.), IEEE Press, 1553～1558

[80] Y.Y.Yao, Information granulation and rough set approximation, International Journal of Intelligent Systems, 2001,Vol. 16, No. 1, 87～104

[81] Y.Y.Yao,Information Granulation and Approximation in a Decision-theoretic Model of Rough Sets, in: Rough-neuro Computing: a Way to Computing with Words, Polkowski , L., Pal, S.K., and Skowron, A. (Eds), Physica-Verlag, Hei-delberg

[82] Y.Y.Yao.and N. Zhong, Granular Computing using Information Tables, in:Data Mining, Rough Sets and Granular Computing, Lin, T.Y., Y.Y.Yao and Zadeh, L.A. (Eds.), Physica-Verlag, Heidelberg,2002, 102～124

[83] Y.Y.Yao. and Yao, J.T., Granular computing as a basis for consistent classification problems, PAKDD Workshop on Fundation of Data Mining, Taipei,Taiwan, May 6-8, 2002, Communications of Institute of Information and Computing Machinery, Taiwan, Vol. 5, No. 2, 101～106

[84] Y.Y.Yao, on modeling data mining with granular computing, proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC 2001), Chicago, USA, October 8-12, 2001, IEEE Computer Society, Los Alamitos, California, 638～643

[85] Y.Y.Yao. and Zhong, N., Potential applications of granular computing in knowledge discovery and data mining, Proceedings of World Multiconference on Systemics, Cybernetics and Informatics, Volume 5, Computer Science and Engineering, Orlando, Florida, USA, July 31 ugust 4, 1999, Torres, M.,Sanchez, B. and Aguilar, J. (Ed.), International Institute of Informatics and Systematics, Orlando, 573～580

[86] Y.Y.Yao, Granular computing using neighborhood systems, Advances in Soft Comp -uting: Engineering Design and Manufacturing, The 3rd On-line World Conference on Soft Computing (WSC3), June 21～30, 1998, Roy, R., Furuhashi,T., and Chawdhry, P.K. (Eds), Springer-Verlag, London, 1999, 539～553

[87] 苗夺谦，王国胤，刘清，林早阳，姚一豫编著. 粒计算：过去、现在与展望. 科学出版社，2007，8

[88] 张钹，张铃. 问题求解理论与应用. 北京：清华大学出版社，1990

[89] 卜东波，白硕，李国杰. 聚类/分类中的粒度原理. 计算机学报，2002，8，810～815

[90] Theresa Beaubouef, Frederick E. Petry ,Gurdial Arora. Information-theoretic measures of uncertainty for rough sets and rough relational databases” .Journal of Information Sciences 109 (1998) 185～195

[91] Theresa Beaubouef, Frederick E. Petry. Intuitionistic Rough Sets for Database Applications. J.F. Peters et al. (Eds.): Transactions on Rough Sets VI, LNCS 4374, 26～30, 2007. Springer-Verlag Berlin Heidelberg 2007

[92] T. Beaubouef, F. Petry, and R. Ladner,“Spatial Data Methods and Vague Regions: A Rough Set Approach,”Applied Soft Computing Journal, vol. 7, January 2007, 425～440

[93] T. Beaubouef and J. Mason, “Why the High Attrition Rate in Computer Science Students: Some Thoughts and Observations,” SIGCSE Bulletin (inroads), Vol. 37, Number 2, June, 2005 pp. 103～106

[94] T. Beaubouef and F. Petry, “Representation of Spatial Data in an OODB Using Rough and Fuzzy Set Modeling,” Soft Computing Journal, vol. 9, No. 5, May 2005, 364～373

[95] T. Beaubouef, R. Ladner, and F. Petry, “Rough Set Spatial Data Modeling for Data Mining,” International Journal of Intelligent Systems, Vol. 19, No. 7, July 2004, pp. 567～584

[96] 胡可云，眭跃飞，陆玉昌，王驹，石纯一等. 多值粗糙集模型. 计算

机科学，2001，28(5):1～4

[97] 曹付元，梁吉业. 基于 SQL 语言的粗糙数据查询. 计算机科学，2004 年 02 期

[98] 郭景峰，李莉，宫继兵. 粗关系数据库中的粗函数依赖研究. 计算机科学，2004 31(9)

[99] 郭景峰，宫继兵，李莉，刘佳. Rough 关系数据库上查询事务处理的研究. 《第二十届全国数据库学术会议》，2003 年，第 20 卷

[100] 张熠，张金城，黄兵. 一种粗糙查询的 SQL 实现新方法. 南京审计学院学报，2007 年 02 期

[101] 王丹，吴孟达，刘银山. 粗糙关系数据库空间结构及其粗糙集模型. 计算机工程与应用 2005(34)

[102] 魏玲玲，邱桃荣，刘萍. 粗关系数据库中的数据更新. 计算机工程，2008, 34(4)

[103] 邱卫根，徐相林. 基于 RST 的粗关系数据库的熵研究. 系统工程与电子技术，2008 30(4)

[104] 邱桃荣，葛寒娟，魏玲玲，徐苏，姚晓昆. 基于相似度的粗关系数据库的近似查询. 计算机工程与应用，2008 44(21)

[105] 马垣著. 非经典关系数据库理论. 北京：清华大学出版社，2005.9

[106] Qiusheng An, Guoyin Wang, Junyi Shen, Jiucheng Xu. Querying data from RRDB based on Rough Sets Theory. The 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC’2003), Chongqing, China, 10, 2003.342～345

[107] Qiusheng An, Junyi Shen. Granular Computing on Functional Dependencies for Information System.RSCTC 2004, LNAI 3066

[108] Qiusheng An, Wenxiu Zhang. The Measures Relationships Study of Three Soft Rules Based on Granular Computing. Rough Sets and Knowledge Technology, LNAI 4062, 2006.7

[109] 安秋生 沈钧毅. 基于信息粒度与 Rough 集的聚类方法研究.《模式识别与人工智能》. 2003,12，第 16 卷第 4 期，412～417

[110] 安秋生 徐久成 王国胤 沈钧毅. 基于粗糙关系数据库的粗糙数据查询.《西安交通大学学报》，2002，第 36 卷第 8 期，859～862

[111] 安秋生 沈钧毅. 信息系统函数依赖的信息颗粒原理与计算. 《西安交通大学学报》，2003, 第 37 卷第 10 期, 999～1002

[112] 安秋生 沈钧毅 王国胤 徐久成. Rough 函数依赖及其推理机制. 《小型微型计算机系统》，2004 第 25 卷第 4 期，638-641, 72～74

[113] 安秋生 徐久成 沈钧毅 王国胤. Rough 关系数据库模型及其关系操作.《计算机科学》，第 29 卷第 7 期，72～74

[114] 安秋生 沈钧毅 王国胤. 基于 RSDA 的 RDB 与 Rough 集关系的研究.《计算机工程与应用》，2002 年第 38 卷第 17 期，21～24

[115] 安秋生 沈钧毅 王国胤. 数据库精确约简合并原理的 Rough 解释.《计算机工程与应用》，2003 年第 39 期卷第 3 期，22～24

[116] 安秋生 沈钧毅，徐久成. 事务工作流与数据库事务模型.《计算机科学》2001.VOL 28 No.5

[117] Qiusheng An, Junyi Shen, An extension of classical functional dependency: rough functional dependency and its inference rules. The 6th International Conference of Electronic Measurement and Instrument (ICEMI'2003), Taiyuan, China, 8,2003,105～108

[118] Qiusheng An, Yusheng Zhang, and WenXiu Zhang,The Study of Rough Relational Database Based on Granular Computing,IEEE International Conference on Granular Computing(GrC 2005)

[119] 安秋生. 基于 GrC 的信息系统软规则及其度量关系的研究.《计算机工程与应用》，2005 第 41 卷第 11 期

[120] 安秋生，张文修. RRDB 与 FRDB 的关系研究. 《计算机工程与应用》，2007 第 43 卷第 6 期

[121] 安秋生，唐淑美. 基于粒计算的 RRDB 的数据查询与函数依赖的研究.《计算机科学》，2005 专刊

[122] Shumei Tang, Qiusheng An.A Systematic Study of the Relationship between RRDB and FRDB.《南昌工程学院学报》，2006 第 25 卷第 2 期

[123] 安秋生. 粗糙函数依赖的近似度量. 《计算机工程与应用》，2009 第 45 卷第 1 期

[124] 萨师煊，王珊. 数据库系统概论（第二版），高等教育出版社，171～174, 1990.5

[125] 王珊，陈红编著，数据库系统原理教程，清华大学出版社，1998.7

[126] 曾黄麟.粗集理论及其应用. 重庆大学出版社，1996,9

[127] Sujeet Shenoi,Austin Melton. Proximity Relations in the Fuzzy Relational Database Model, Fuzzy Sets and Systems 100 Supplement(1999)51-62,North-Holland

[128] Therasea Beauboyef, Frederick E.Petry. Rough Functional Dependencies. IKE’04 International Conference

[129] Sakai H.An enhancement of a procedure for checking dependencies of attributes in a table with non-deterministic information.Proceedings of International Workshop on Rough Set Theory and Granular Computing. Bulletin of International Rough Set Society. Matsue, Shimane, Japan May 20-22,2001.81~87

[130] Hiroshi Sakai and Akimichi Okuma, Basic Algorithms and Tools for Rough Non-deterministic Information Analysis. In: J.F. Peters et al. (Eds.): Transactions on Rough Sets I, LNCS 3100, 209～231, 2004. © Springer-Verlag Berlin Heidelberg 2004

[131] 王能斌. 数据库系统原理. 电子工业出版社，2000 年 1 月第 1 版，116～118

[132] 陶影. 模糊查询和模糊数据在数据库中的应用. 黄金学报，2001 年 9 月第 3 卷第 3 期，216～218

[133] 鹤荣育. 基于概率的数据库模糊查询. 微电子学与计算机，1994 年第 6 期，42～28

[134] 周少泉，丁立新，唐新华，张键. 一类模糊函数依赖. 武汉大学学报，1997 年 2 月第 43 卷第 1 期

[135] B.P.Buckles, F.E.Petry,A fuzzy respresentation of data for relational databases,

Fuzzy Sets and Systems 7(3)(1982)213～226

[136] Mustafa Ilker Sozat,Adnan Yazici.A complete axiomatization for fuzzy and multivalued dependencies in fuzzy database relations.Fuzzy Sets Systems 117, 161～181 , 2001

[137] Z.M.Ma,W.J.Zhang,F.Mili.Fuzzy Data Compress Based on Data Dependencies, International Journal of Intelligent Systems,Vol, 17, 409～426(2002)

[138] ARABIE, P. and HUBERT, L.J. 1996. An overview of combinatorial data analysis, in: Arabie, P., Hubert, L.J., and Soete, G.D. (Eds.) Clustering and Classification, 5-63, World Scientific Publishing Co., NJ

[139] DUDA, R. and HART, P. 1973. Pattern Classification and Scene Analysis. John Wiley & Sons, New York, NY

[140] GERSHO, A. and GRAY, R. M. 1992. Vector Quantization and Signal Compression. Communications and Information Theory. Kluwer Academic Publishers, Norwell, MA

[141] Pavel Berkhin, Survey of Clustering Data Mining Techniques,NEC Research Institute, 1997～2000

[142] Shoji Hirano,Shusaku Tsumoto Tomohiro Okuzaki,Yutaka Hata. A Clustering Method for Nominal and Numerical Data Based on Rough Set Theory. Bulletin of International Rough Set Society Proceedings of RSTGC 2001, Volume 5, No.1/2,211～216

[143] M .肯德尔著. 多元分析. 北京：科学出版社，1983-7，第 1 版，215～216

[144] 苗夺谦，范世栋. 知识的粒度计算及其应用. 系统工程理论与实践. 2002, 1, 48～56

[145] Aditya N.Saharia,Terence M.Barron.Approximate dependencies in database systems. Decision Support Systems 13(1995)335～347

[146] ZHEXUE HUANG.Extensions to the K-means Algorithm for Clustering Large Data Sets with categorical Values.Data Mining and Knowledge Discovery 2, 283～304 (1998)

[147] Christopher J.Matheus,Philip K.Chan, and Gregory Piatetsky-Shapiro. Systems for Knowledge Discovery in Databases.IEEE Transactions on Knowledge and Data Enginee -ring,Vol,5,No.6 December 1993, 903～913

[148] Eric Louie and T.Y.Lin, A data Mining Approach using Machine-Oriented Modeling: Finding Association Rules using Canonical Names. In Data Mining and Knowledge Discovery :Theory,Tools, and Technology 2,Belur V. Dasarathy,Editor, Proceedings of SPIE Vol.4057(2000).148～154

[149] T.Y.Lin.Data Modeling for Data Mining.Data Mining and Knowledge Discovery :Theory,Tools and Technology IV,Belur V.Dasarathy, Editor. Proceedings of SPIE Vol.4730(2002)[A].138～145

[150] CODD. E. F. Recent investigations in relational database systems. Information Processing 74, North-Holland Pub. Co., Amsterdam, 1974, pp. 1017～1021

[151] Shi Bo-le,Ding Bao-Kang. Database System Tutorial (Second Edition) [M]. Beijing: Academic Education Press .2003, 8

[152] M.J.Fischer, “Notes on Functional Dependencies and partitions(CPSC 437b: Introduction to Databases).” February 28, 2003, Handout #10

[153] Philip A. Bernstein, Nathan Goodman, WHAT DOES BOYCE-CODD NORMAL FORM DO, Sixth International Conference on Very Large Data Bases[M], October 1-3, 1980,Montreal, Quebec, Canada, Proceedings. IEEE-CS, 1980, IEEE Catalog Number 80CH1534-7C, pp 245～259

[154] T.Y.Lin,Eric Louie,Data Mining Using Granular Computing:Fast Algorithms for Finding Association Rules,In: Data Mining, Rough Sets and Granular Computing, T.Y.Lin,Y.Y.Yao,and L.A.Zadeh,Eds. Physics, Heidelberg, 2003, 22～42

[155] F. Berzal, J.C. Cubero,F. Cuenca, J.M. Medina.Relational decomposition through partial functional dependencies, Data & knowledge Engineering 43 (2002) 207～234

[156] Chris Giannella, Edward Robertson.On approximation measures for functional dependencies, Information Systems 29 (2004) 483～507